

Conexión a una base de datos
mediante phpMyAdmin dentro de
un hosting en HOSTINGER.ES

Y

Conexión a una base de datos
desde una página web mediante
código PHP

Índice

ENUNCIADO	3
BASE DE DATOS	3
Modelo Entidad Relación	3
Modelo Relacional	3
Tablas	4
Script SQL completo	5
PASOS PARA UTILIZAR PHPMYADMIN	9

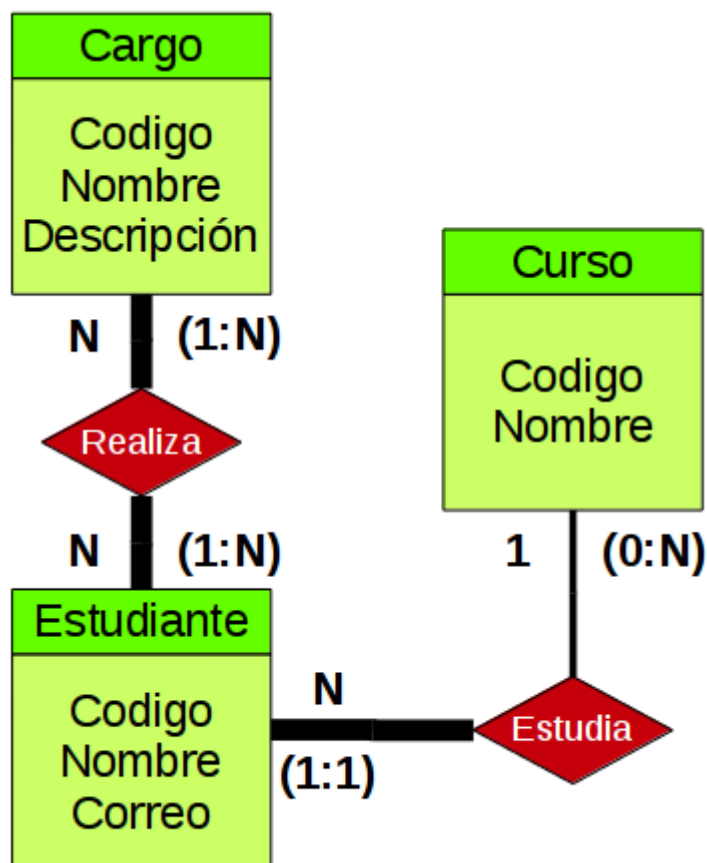
ENUNCIADO

En ToxoSoftware necesitamos documentar cómo se realiza una conexión a una base de datos mediante PHP, para ello utilizaremos un Modelo Entidad Relación para crear una base de datos y así probar que phpMyAdmin y el servidor MYSQL funcionan correctamente.

BASE DE DATOS

Modelo Entidad Relación

Para el modelo Entidad Relación propongo guardar los cargos actuales de cada uno de los integrantes así como también el curso que están estudiando en estos momentos.



Modelo Relacional

Referencias:

Clave Primaria ~~Clave Secundaria~~ Clave Primaria Y Secundaria

Relaciones:

Cargo (codigo, nombre, descripción)

Curso (codigo, nombre)

Estudiante (codigo, nombre, correo, ~~codigo_curse~~)
codigo_curso → Curso (codigo)

Realiza (~~codigo_cargo~~, ~~codigo_estudiante~~)
codigo_cargo → Cargo (codigo)
codigo_estudiante → Estudiante (codigo)

Tablas

Tabla	Campo	Tipo	Restricción	Referencia	Requerido
Cargo	cod_cargo	tinyint	PK		Si
	nom_cargo	Varchar(50)			Si
	desc_cargo	Varchar(200)			No
Estudiante	cod_est	tinyint	PK		Si
	nom_est	Varchar(50)			Si
	correo_est	Varchar(50)	UQ (trigger)		No
	cod_cur	tinyint	FK	Curso	SI
Curso	cod_cur	tinyint	PK		Si
	nom_cur	Char(8)			Si
Realiza	cod_cargo	tinyint	PK / FK	Cargo	Si
	cod_est	tinyint	PK / FK	Estudiante	Si

Script SQL completo

NOTA: Al momento de crear el contenido dentro de la base de datos, habrá que quitarle las 3 primeras sentencias: drop, create y use; porque en el hosting ya se realizan automáticamente. Dejo el script completo porque si alguien quiere hacer pruebas sobre la base de datos en un servidor MYSQL local, tiene todo lo necesario para ejecutar dicho script.

```
-- Con esta línea se borra la base de datos si existiera
-- está creada para que se pueda ejecutar el script
-- tantas veces como el usuario quiera
drop database if exists toxosoftware_db;
```

```
-- Creación de la base de datos con caracteres utf8
-- así se pueden meter tildes y otros no dando error
-- al momento de las inserciones o consultas
create database toxosoftware_db
    default character set utf8
    default collate utf8_general_ci;
```

```
-- Ponemos el foco en la base de datos creada
use toxosoftware_db;
```

```
-----
-- A partir de aquí vamos a copiar y pegar el script
-- dentro del cuadro SQL de PHPMyAdmin
-----
```

```
-- Tabla CARGO
create table cargo (
    cod_cargo          tinyint          auto_increment,
    nom_cargo          varchar(50)      not null,
    desc_cargo         varchar(200),
    primary key (cod_cargo)
);
```

```
-- Tabla CURSO
create table curso (
    cod_cur            tinyint          auto_increment,
    nom_cur            char(8)          not null,
    primary key (cod_cur)
);
```

```
-- Tabla ESTUDIANTE
create table estudiante (
    cod_est            tinyint          auto_increment,
    nom_est            varchar(50)      not null,
    correo_est        varchar(50),
    cod_cur            tinyint          not null,
```

```

        primary key (cod_est),
        constraint fk_estudiante_curso foreign key (cod_cur)
            references curso (cod_cur)
    );

-- Tabla REALIZA (relación entre Estudiante y Cargo)
create table realiza (
    cod_cargo          tinyint,
    cod_est            tinyint,
    primary key (cod_cargo, cod_est),
    constraint fk_realiza_cargo foreign key (cod_cargo)
        references cargo (cod_cargo),
    constraint fk_realiza_estudiante foreign key (cod_est)
        references estudiante (cod_est)
);

-- Insertar datos en la tabla CARGO
insert into cargo values
    (1, 'Jefe Proyecto','Distribuye el trabajo'),
    (2, 'Analista Funcional','Recojen información del cliente para
la puesta en marcha del proyecto'),
    (3, 'Diseñador','En función de los datos recaudados por los
analistas funcionales, crearán una maqueta para esneñarle al
cliente'),
    (4, 'Fotógrafo','En función de los datos recaudados por los
analistas funcionales, tomarán fotos de paisajes y zonas verdes como
se les solicite'),
    (5, 'Programador','Crearán la página web final con la que
interactuarán los clientes tomando los datos de fotografía y de
diseño'),
    (6, 'Tester','Revisan y comprueban que la página web funcione
correctamente, redactarán manuales de explicación del funcionamiento
de la página web para los clientes');

-- Insertar datos en la tabla CURSO
insert into curso values
    (1, 'ASIR-1'),
    (2, 'ASIR-2');

-- Insertar datos en la tabla ESTUDIANTE (cuidado con las FK)
insert into estudiante values
    (1, 'Félix Pardal
González','a14felixpg@iessanclemente.net',1),
    (2, 'Pablo Pérez García','a14pablogp@iessanclemente.net',1),
    (3, 'Pablo Otero García','a14pabloog@iessanclemente.net',1),
    (4, 'Daniel Rey Castro','a15danielrc@iessanclemente.net',1),
    (5, 'Daniel Herranz
Chenlo','a15danielhc@iessanclemente.net',2),
    (6, 'Mario Marcelo Michelotti
Pan','a15mariom@iessanclemente.net',2);

-- Insertar datos en la tabla REALIZA (cuidado con ambas FKs)
insert into realiza values
    (1,1),
    (2,2),

```

```

(2,3),
(3,1),
(3,3),
(4,1),
(4,2),
(4,4),
(5,5),
(5,6),
(6,4),
(6,6);

-- Esta vista está para que la anterior tabla sea más legible
-- Con una select sobre esta vista sabremos
-- el nombre y el cargo del estudiante
create view vista_realiza as
    select e.nom_est, c.nom_cargo from realiza as r
        join estudiante as e using (cod_est)
        join cargo as c using (cod_cargo);

-- Función general para saber si un correo electrónico ya se ha
-- devuelve NULL si no encuentra nada
delimiter @@
create function fn_estudiante_correo (recibe_correo varchar(50))
returns text begin
    declare envia_correo varchar(50);
    select correo_est into envia_correo from estudiante
        where correo_est like recibe_correo;
    return envia_correo;
end@@
delimiter ;

-----
-- Nota sobre estos triggers, como no existe una función para
-- realizarlo, se deben comprobar los registros manualmente,
-- y como se permite que el campo quede en NULL, no se le puede
-- dar la instrucción de que el requerimiento sea UNIQUE porque
-- entonces sólo podría haber 1 único registro NULL.
-----

-- 1er Trigger, para antes de insertar datos en la tabla
-- ESTUDIANTE, si ya existe un correo detiene la transacción
-- y muestra un error por pantalla
delimiter @@
create trigger tr_estudiante_insertar_correo before insert on
estudiante
for each row begin
    if new.correo_est = (select
fn_estudiante_correo(new.correo_est)) then
        signal sqlstate '99999' set message_text = 'No che puede
@@@...';
    end if;
end@@

-- 2do Trigger, para antes de actualizar datos en la tabla
-- ESTUDIANTE, si ya existe un correo detiene la transacción

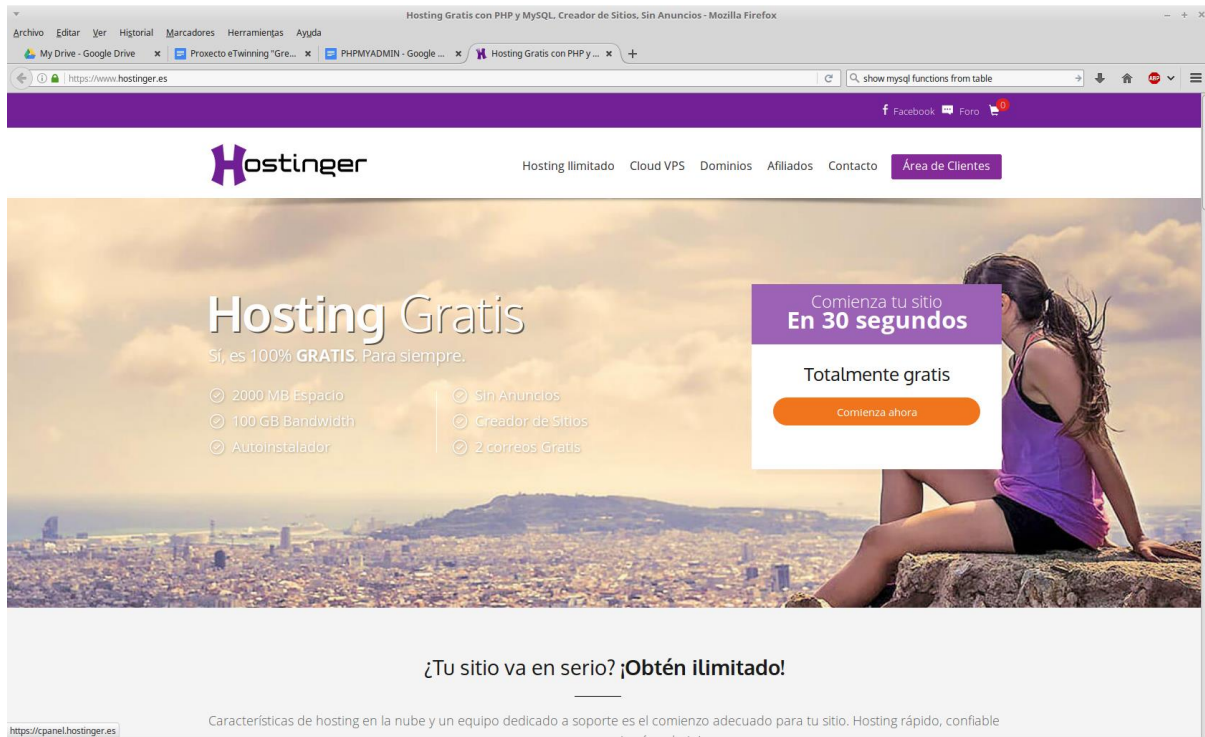
```

```
-- y muestra un error por pantalla
create trigger tr_estudiante_actualizar_correo before update on
estudiante
for each row begin
    if new.correo_est = (select
fn_estudiante_correo(new.correo_est)) then
        signal sqlstate '99999' set message_text = 'No che puede
@@...';
    end if;
end@@

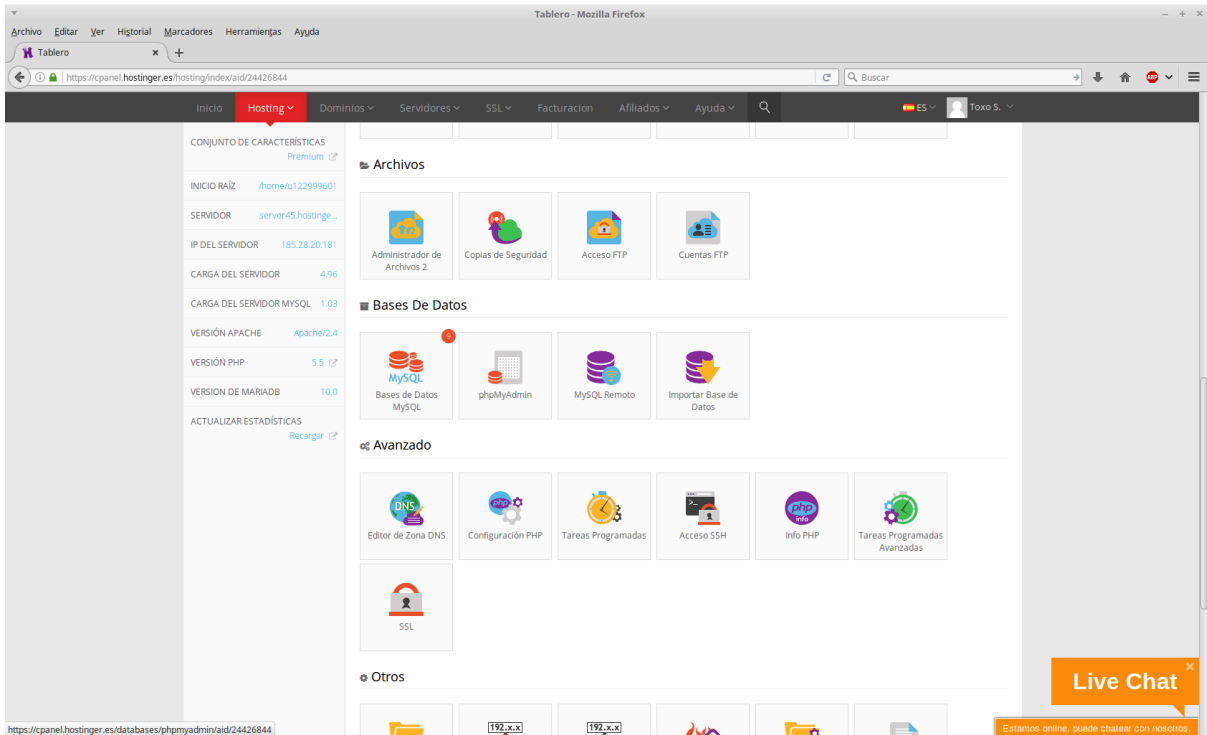
delimiter ;
```


PASOS PARA UTILIZAR PHPMYADMIN

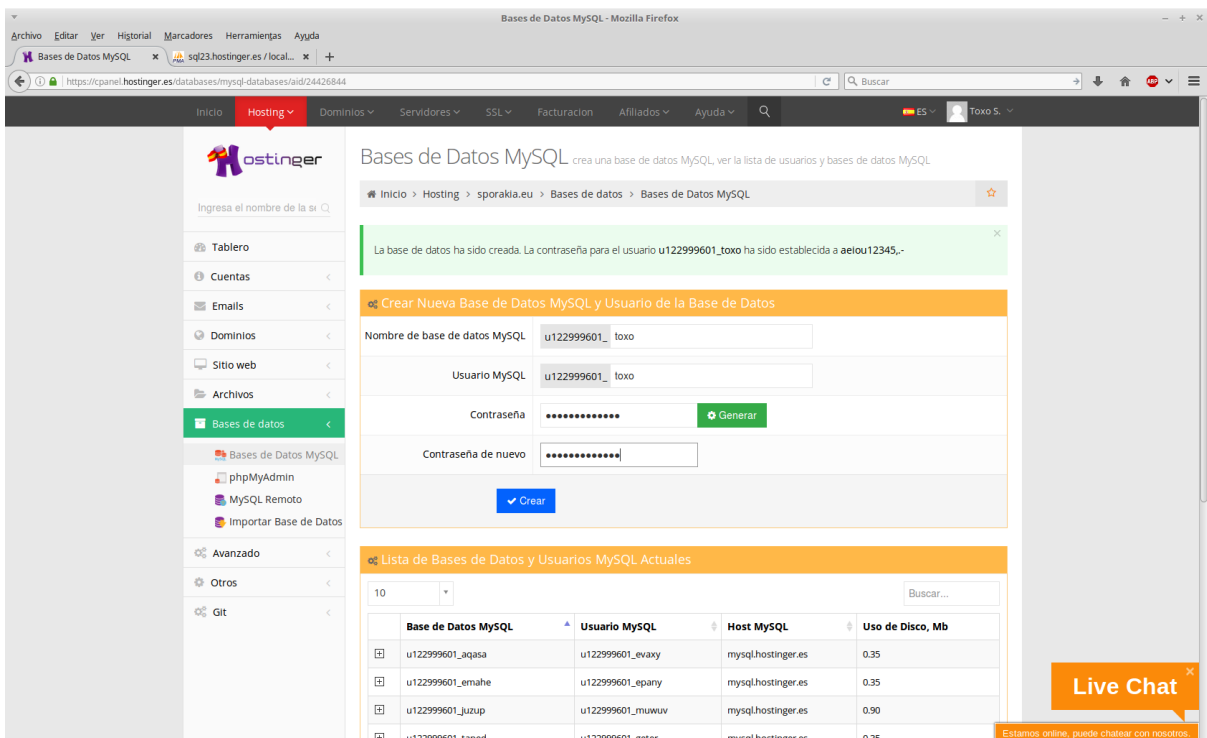
Conectarse a Hostinger mediante el área de clientes



Ir al apartado de Bases de Datos e ingresar en MySQL, vamos a crear la base de datos

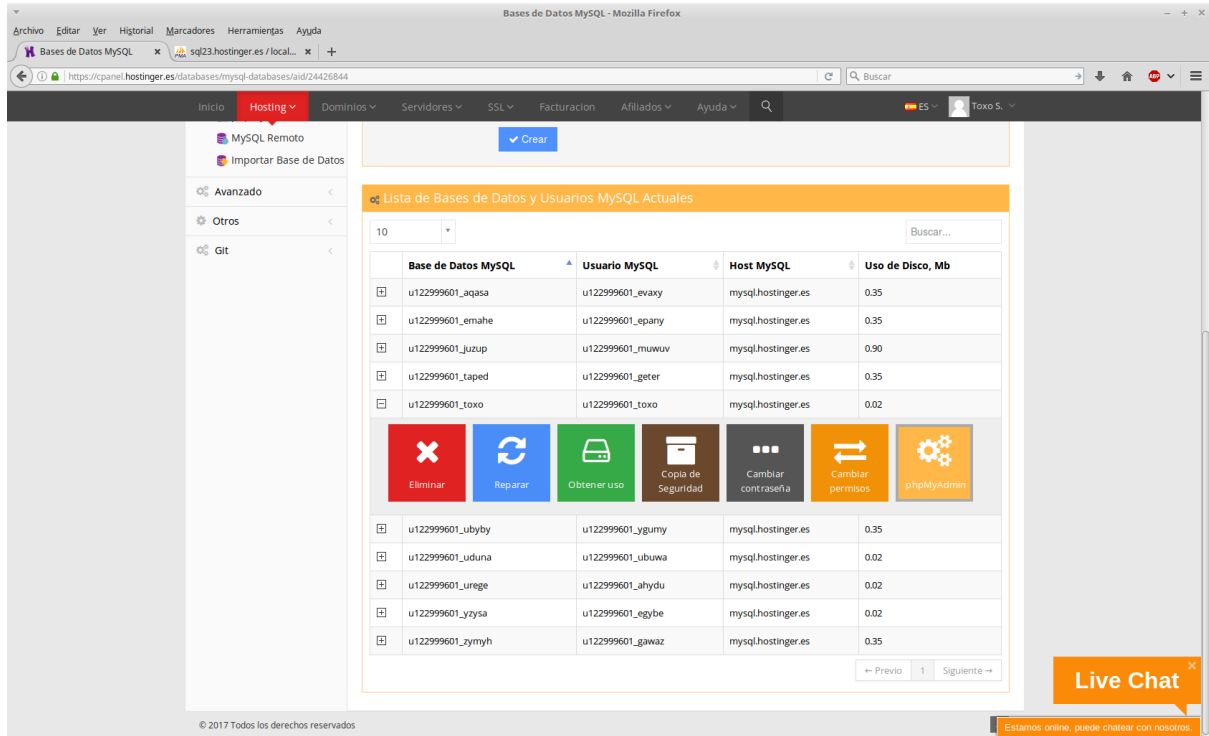


Ingresamos una terminación para la base de datos y para el usuario, elegí el mismo nombre pues en este caso me es más cómodo, pero nunca se debe de hacer. También hay que ingresar una contraseña, en este caso los datos para esta base de datos son:

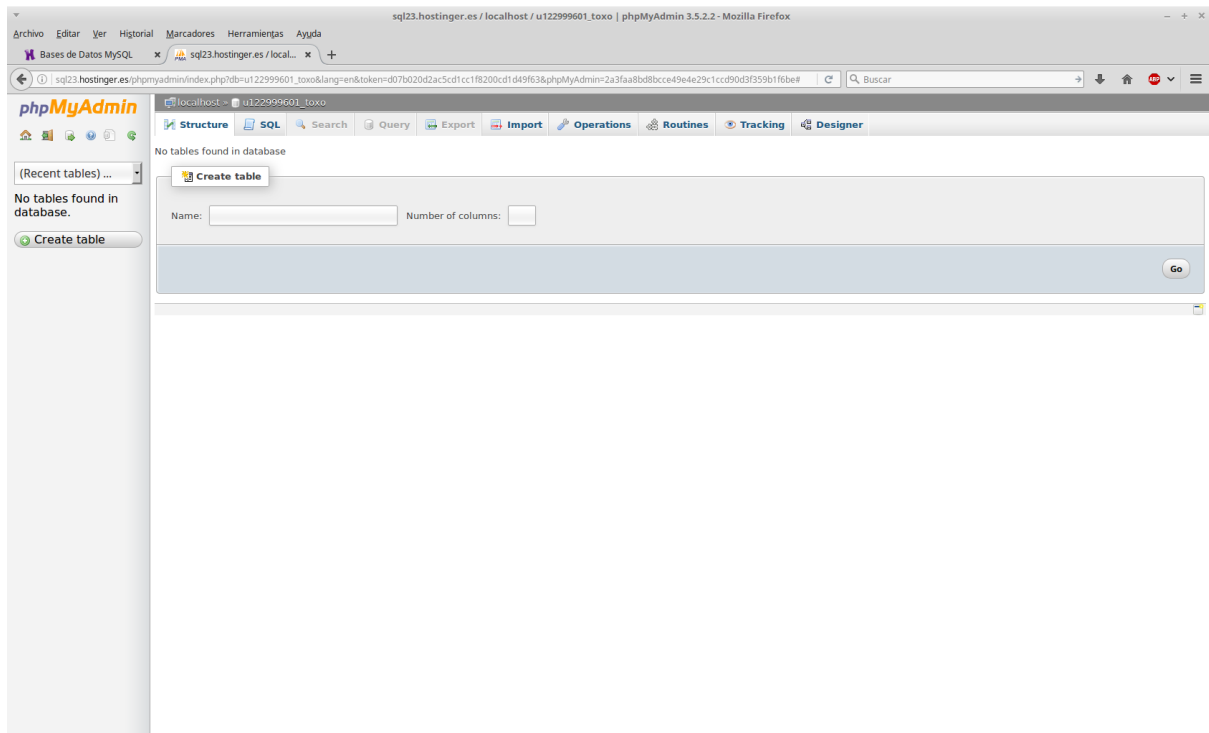


Luego de creada, más abajo ya debería aparecer, la buscamos y le damos un click para que nos muestre unas opciones. Las opciones de

Cambiar contraseña y cambiar permisos hablan por si mismas. Vamos a hacer click sobre los engranes con la leyenda phpMyAdmin.

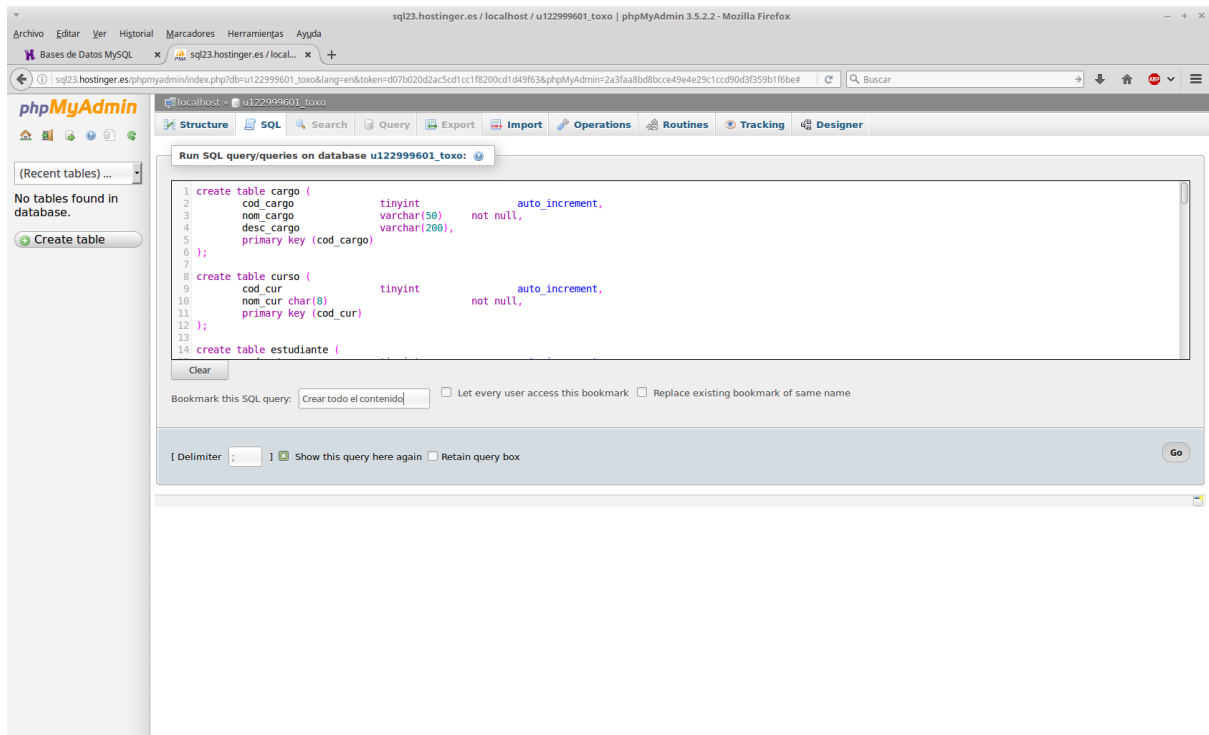


Automáticamente nos abre una nueva pestaña con la interface de phpMyAdmin.

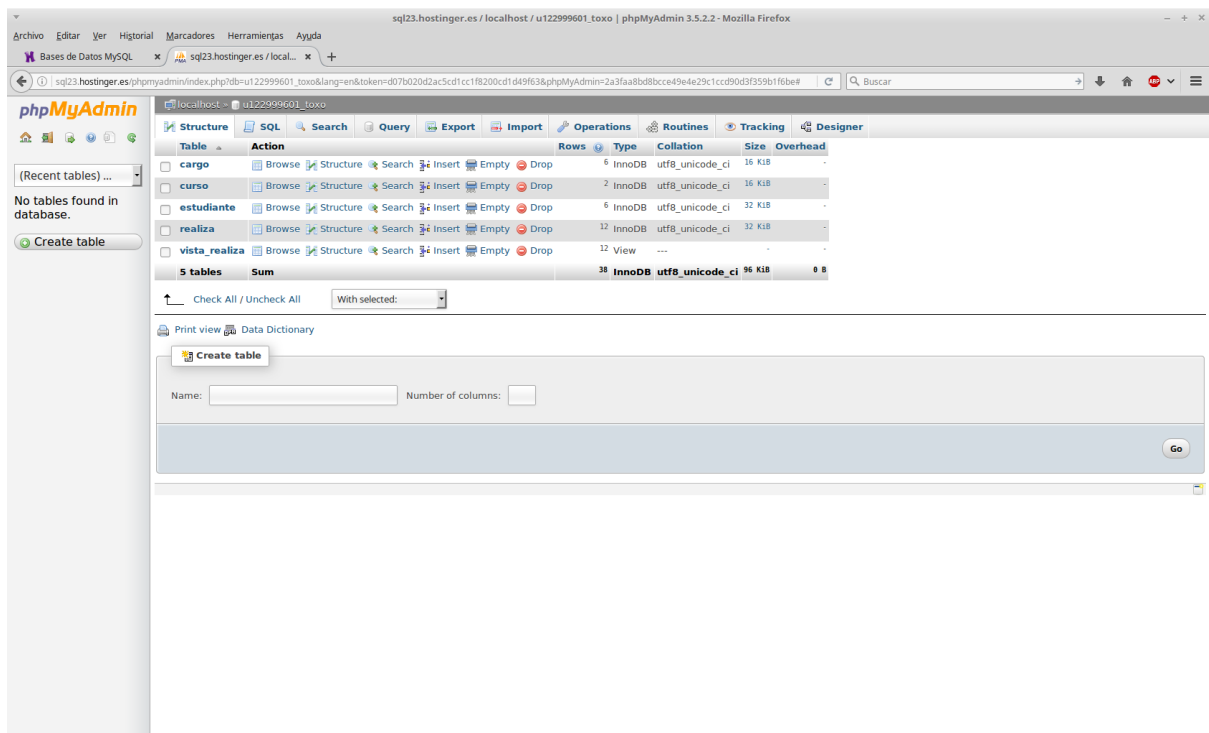


Si hacemos click en el botón de arriba con nombre SQL, nos aparece un recuadro para insertar texto y otro botón más abajo que pone GO, aquí vamos a meter el script sql de arriba para probar su

funcionamiento (desde que comienza a crear las tablas), y le vamos a hacer click en GO.

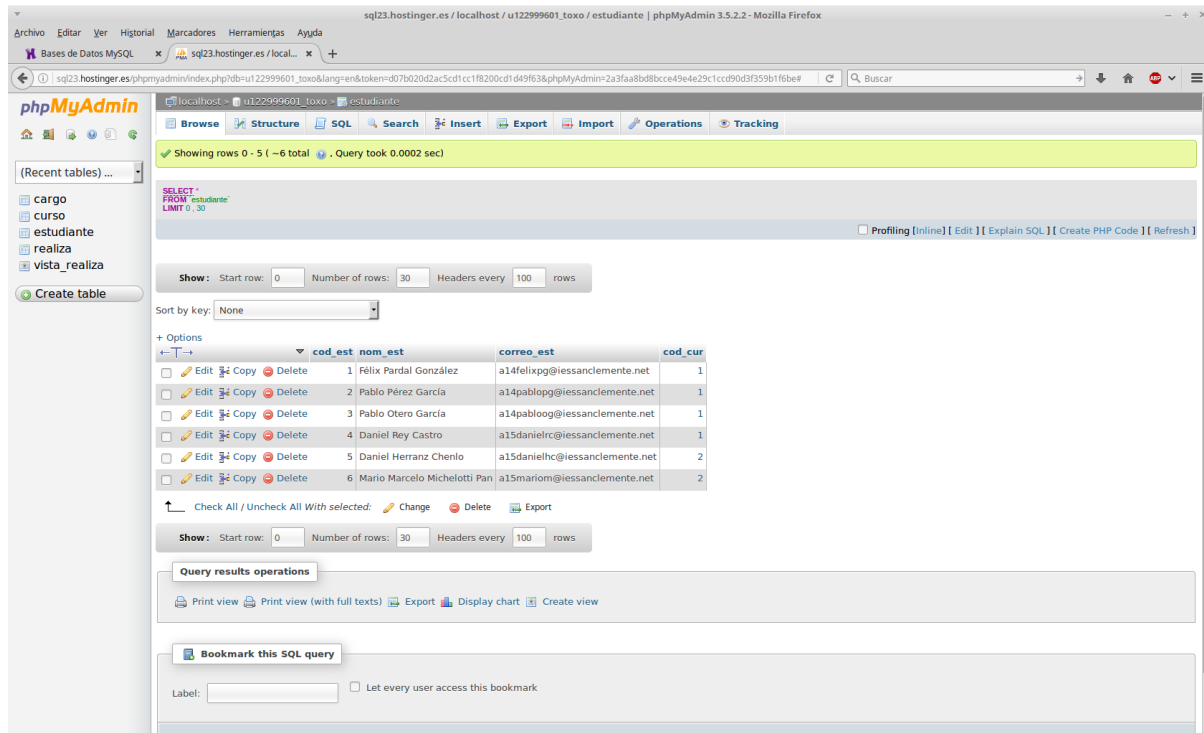


Para ver un mejor resultado, hacemos click en el botón que dice STRUCTURE, así nos mostrará todas las tablas.



Debajo del logo de phpMyAdmin aparece un botón verde para refrescar, pulsándole nos aparecen las tablas que se crearon a la izquierda

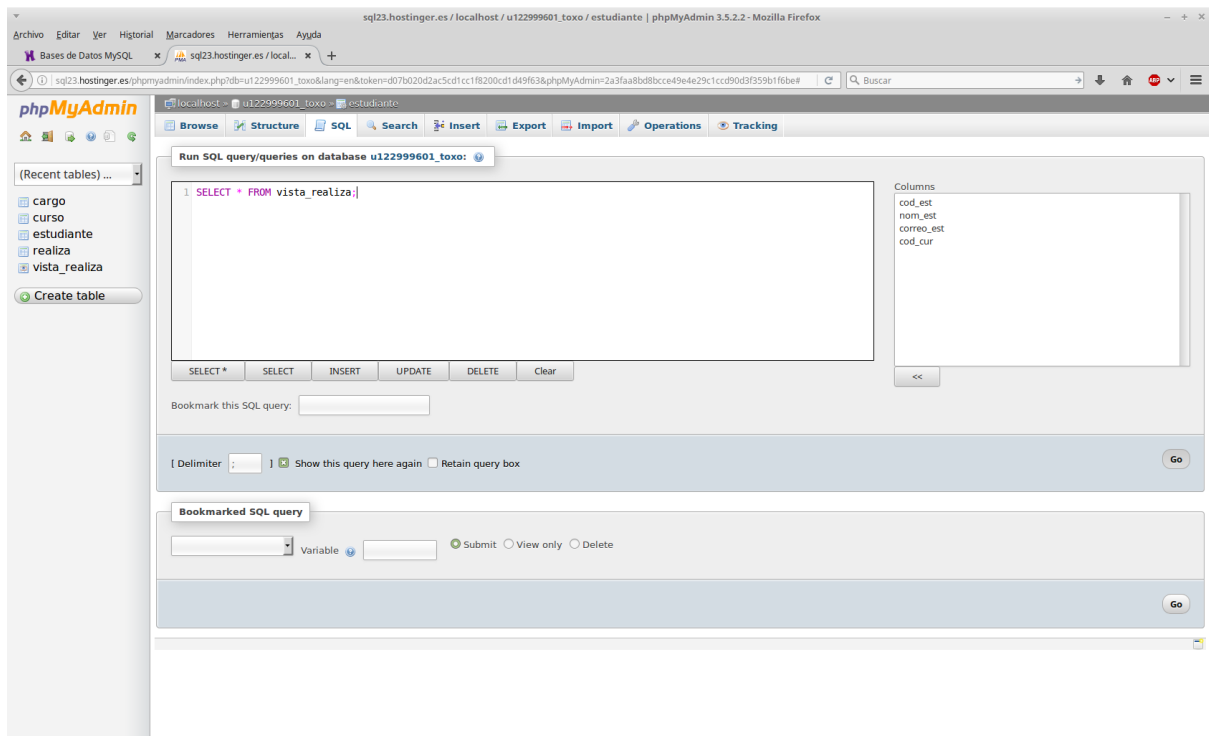
para tener un acceso más directo a las mismas, haciéndole click sobre la tabla ESTUDIANTE (por ejemplo), ya nos muestra su contenido.



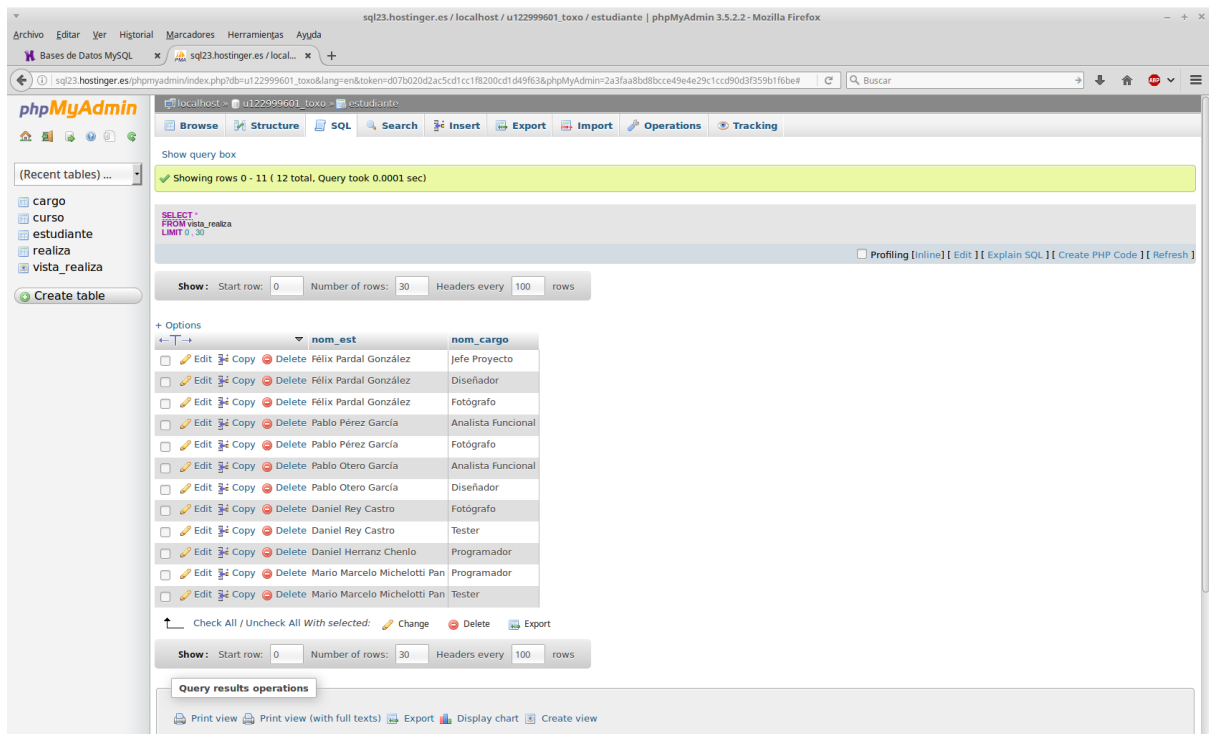
The screenshot shows the phpMyAdmin interface for a MySQL database. The 'estudiante' table is selected, and its contents are displayed in a table format. The table has columns: 'cod_est', 'nom_est', 'correo_est', and 'cod_cur'. The data is as follows:

cod_est	nom_est	correo_est	cod_cur
1	Félix Pardal González	a14felixpg@iessanclamente.net	1
2	Pablo Pérez García	a14pablopg@iessanclamente.net	1
3	Pablo Otero García	a14pabloog@iessanclamente.net	1
4	Daniel Rey Castro	a15danielrc@iessanclamente.net	1
5	Daniel Herranz Chenlo	a15danielhc@iessanclamente.net	2
6	Mario Marcelo Michelotti Pan	a15mariom@iessanclamente.net	2

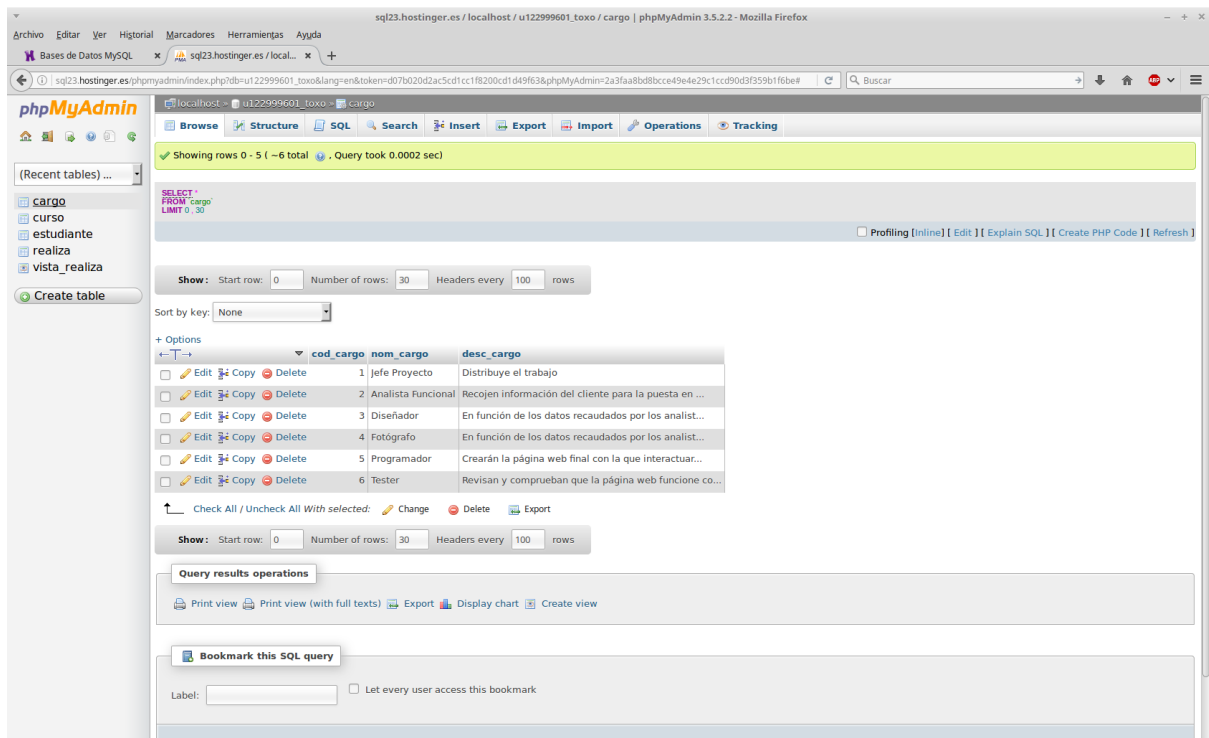
Vamos a probar una consulta de contenido sobre la vista VISTA_REALIZA, clicamos en SQL e ingresamos la consulta:
Select * from vista_realiza;
Y le damos al botón GO.



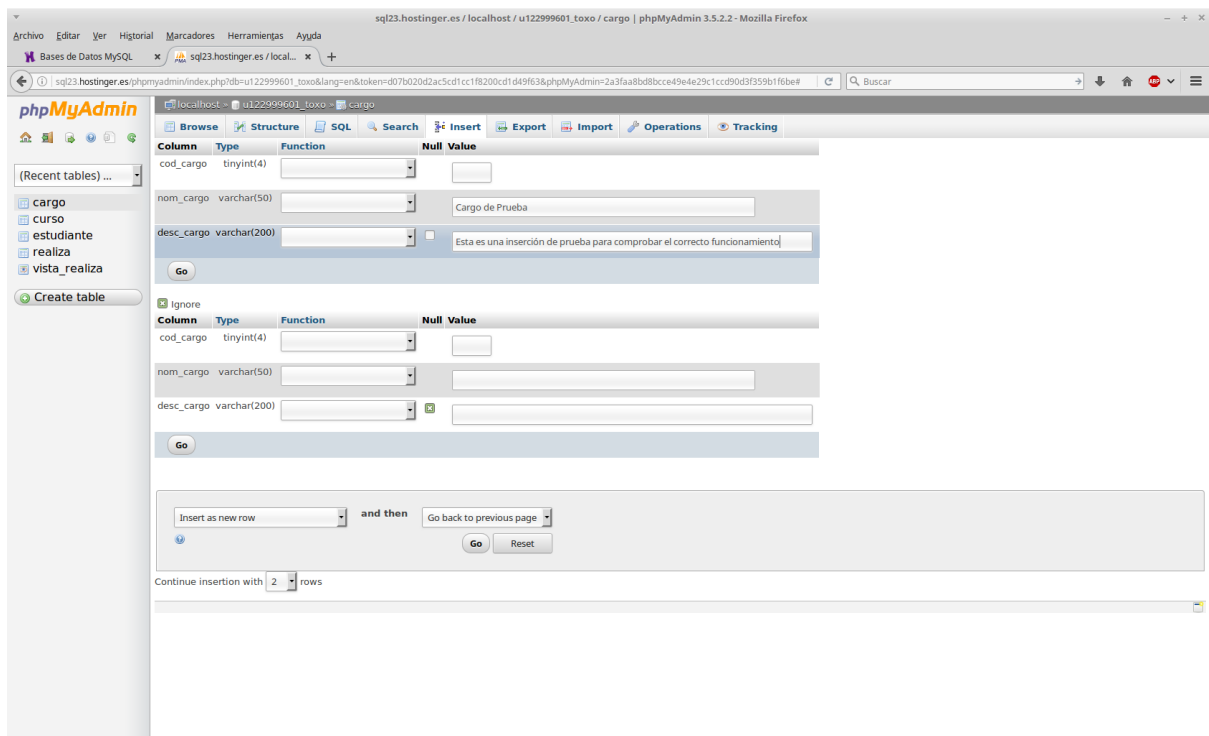
Nos muestra todo su contenido.



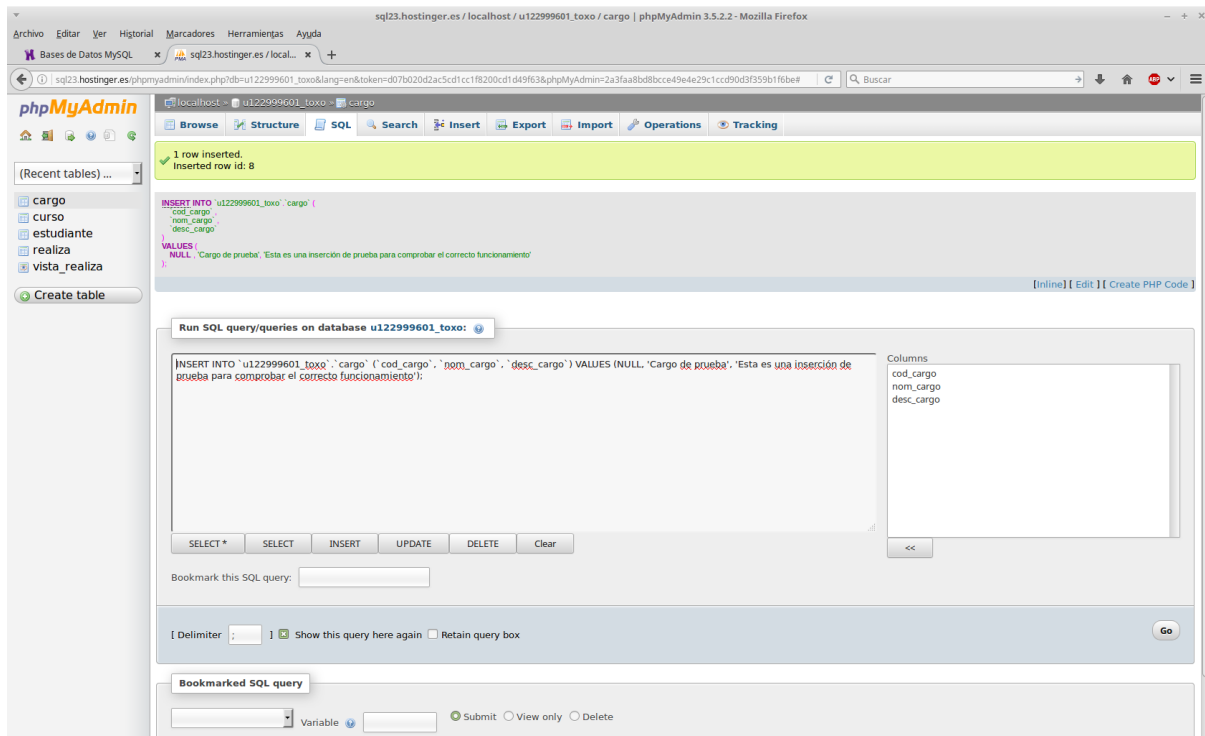
Ahora veamos otro contenido, el de la tabla CARGO, hacemos click sobre su nombre que aparece a la izquierda.



También se puede insertar datos en una tabla mediante la interfaz web, clicamos en el botón de INSERT y veamos, cuidado porque va a insertar sobre la última tabla seleccionada (CARGO en este caso). Ingresamos a modo de prueba un nombre para el nuevo cargo y una descripción, también como modo de prueba vamos a dejar en blanco el campo de COD_CARGO para verificar que el auto_increment sí funciona. Al finalizar le damos click a GO.



phpMyAdmin nos mostrará la consulta que hemos hecho en modo texto, y arriba del todo un mensaje notificando si se ha realizado correctamente o si hubo algún fallo. Aquí informa que todo ha ido bien.



Clicamos sobre la tabla CARGO corroboramos. Efectivamente los datos fueron insertados y (como antes había hecho otra prueba y la borré), nos puso un autoincremental.

The screenshot shows the phpMyAdmin interface with a query result for the 'cargo' table. The query executed is `SELECT FROM cargo LIMIT 0, 50`. The result shows 8 rows of data with columns `cod_cargo`, `nom_cargo`, and `desc_cargo`.

cod_cargo	nom_cargo	desc_cargo
1	Jefe Proyecto	Distribuye el trabajo
2	Analista Funcional	Recojen información del cliente para la puesta en ...
3	Diseñador	En función de los datos recaudados por los analist...
4	Fotógrafo	En función de los datos recaudados por los analist...
5	Programador	Crearán la página web final con la que interactuar...
6	Tester	Revisan y comprueban que la página web funcione co...
8	Cargo de prueba	Esta es una inserción de prueba para comprobar el ...

Al clicar sobre el botón CREATE TABLE, nos aparece una especie de tabla para rellenar y seleccionar opciones, aquí podemos crear nuevas tablas y configurar las propiedades de cada campo. Voy a ingresar 3 campos de diferente tipo y valor.

The screenshot shows the 'CREATE TABLE' structure configuration screen in phpMyAdmin. The table name is 'Nueva Tabla' and it has 1 column(s). The structure is defined as follows:

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I	Comments	MIME type	Browser transform
Campo1	TINYINT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>			
Campo2	VARCHAR	200	As defined: Valor por defecto			<input type="checkbox"/>	UNIQUE	<input type="checkbox"/>			
Campo3	DATE		None			<input type="checkbox"/>	--	<input type="checkbox"/>			
	INT		None			<input type="checkbox"/>	--	<input type="checkbox"/>			

Additional settings: Storage Engine: InnoDB, Collation: [empty].

Al darle a GO veremos la tabla creada correctamente, luego le hacemos click sobre Nueva Tabla a la izquierda (ese es el nombre que le puse arriba), le hacemos click a STRUCTURE y veremos los campos y sus correspondientes características.

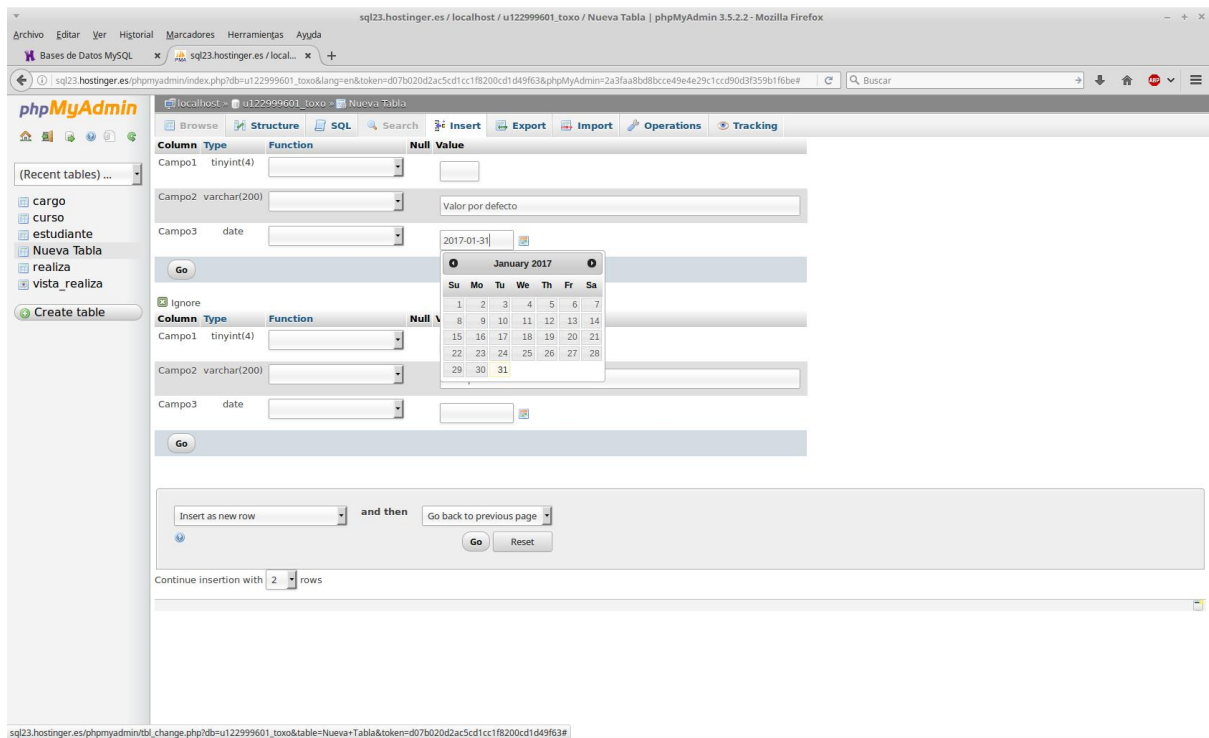
The screenshot shows the phpMyAdmin interface for a table named 'Nueva Tabla'. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	Campo1	tinyint(4)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	Campo2	varchar(200)	utf8_unicode_ci		No	Valor por defecto		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	Campo3	date			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

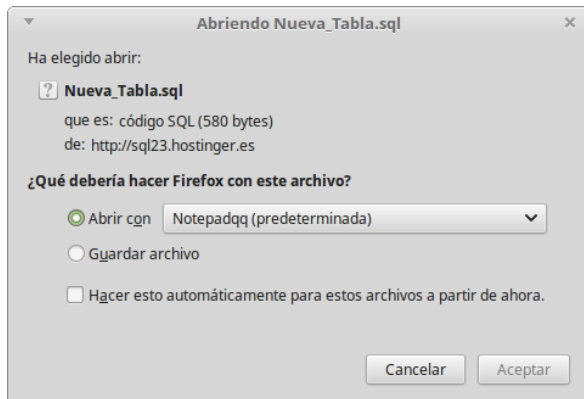
Below the table structure, there is an 'Information' section with the following data:

Space usage	Row Statistics
Data 15 KIB	Format Compact
Index 15 KIB	Collation utf8_unicode_ci
Total 32 KIB	Next autoindex 1
	Creation Jan 31, 2017 at 07:18 PM

Vamos de nuevo a INSERT, ahora vemos un par de cositas nuevas que antes no había porque no se habían configurado de esa forma. En el campo CAMPO2 se había dicho que tuviese un valor por defecto con el texto: "Valor por defecto"... ahí está. Y en el CAMPO3 con un tipo de valor DATE, si le hacemos click al botoncito a su derecha nos desplegará un almanaque para seleccionar la fecha.



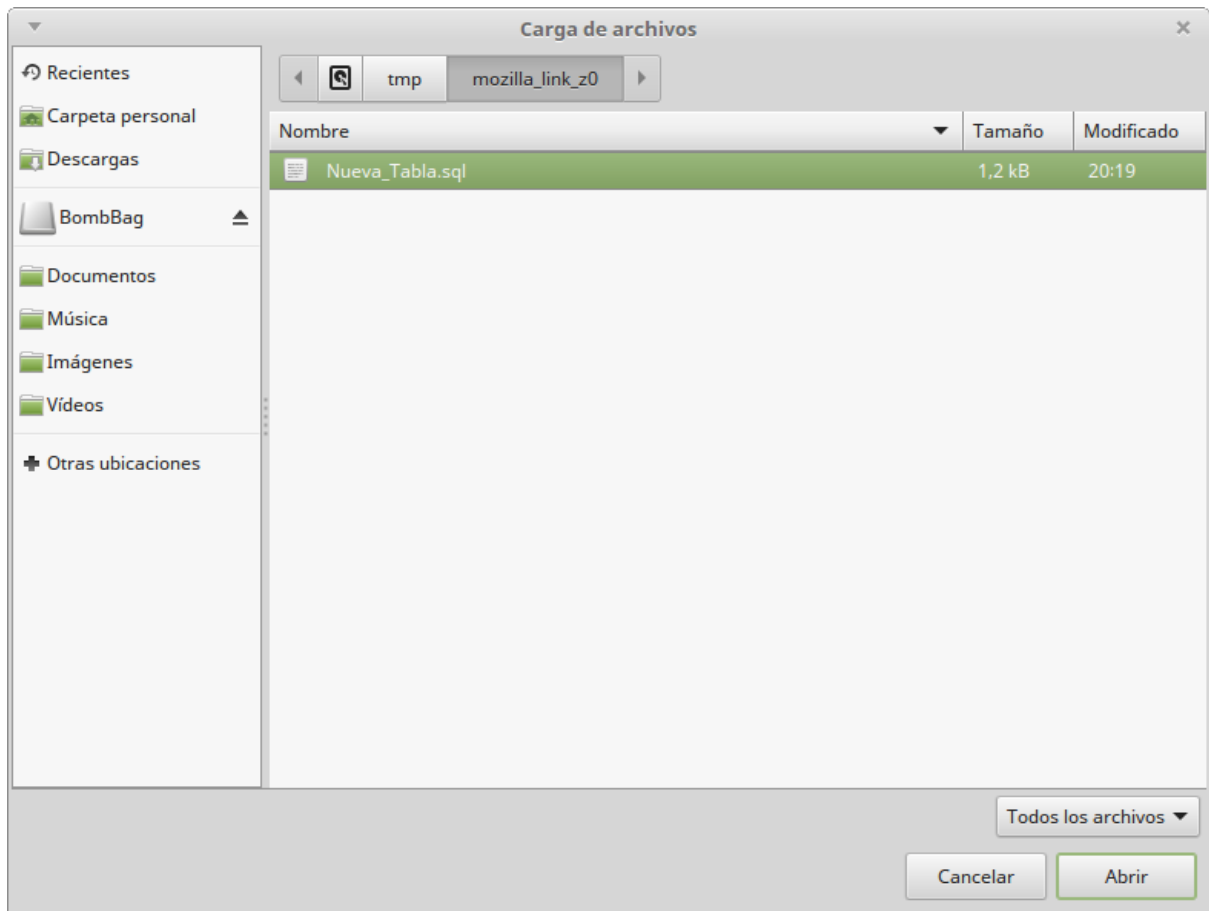
Vamos al siguiente botón, EXPORT, donde podremos exportar la tabla, le damos al botón y luego a GO, nos aparece para descargar la tabla.



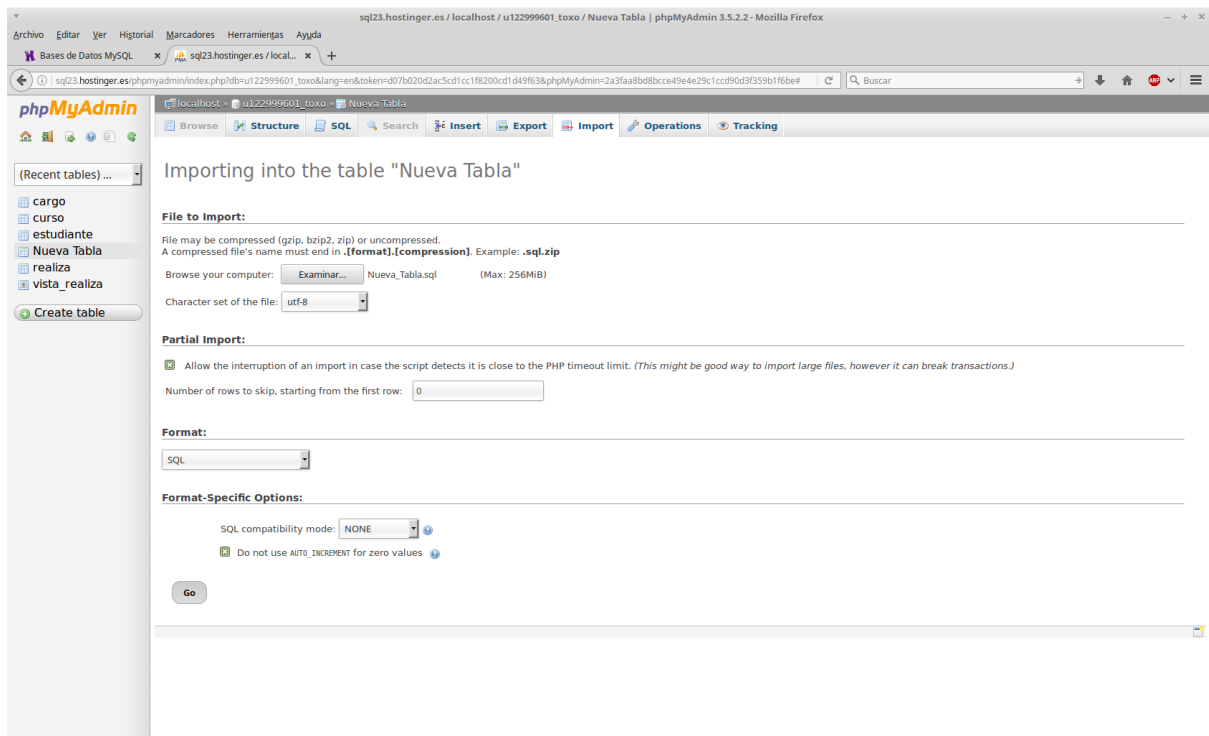
La abrimos y efectivamente aparecen las sentencias sql para crear la tabla.

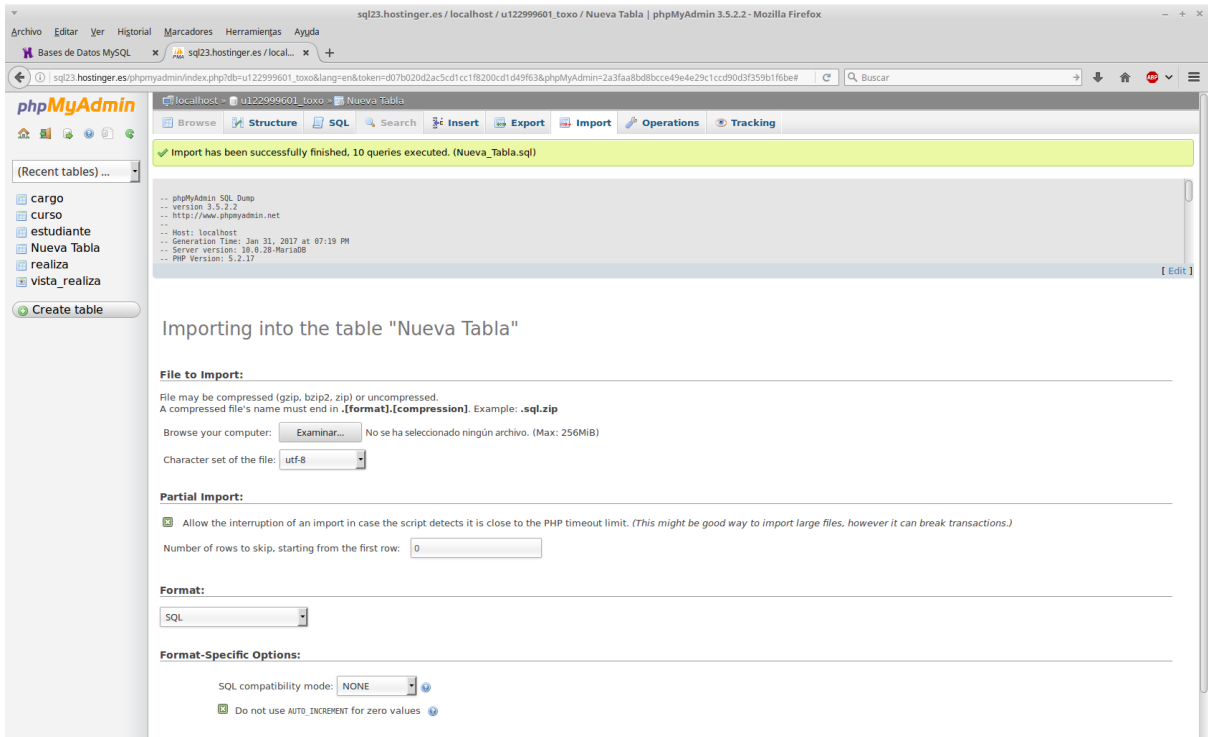
```
Nueva_Tabla.sql (/tmp/mozilla_link_x0) - Notepadqq
File Edit Search View Encoding Language Settings Run Window ?
Nueva_Tabla.sql
1
2 -- phpMyAdmin SQL Dump
3 -- version 3.5.2.2
4 -- http://www.phpmyadmin.net
5 --
6 -- Host: localhost
7 -- Generation Time: Jan 31, 2017 at 07:19 PM
8 -- Server version: 10.0.28-MariaDB
9 -- PHP Version: 5.2.17
10
11 SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
12 SET time_zone = "+00:00";
13
14
15 /*140101 SET @OLD_CHARACTER_SET_CLIENT=@CHARACTER_SET_CLIENT */;
16 /*140101 SET @OLD_CHARACTER_SET_RESULTS=@CHARACTER_SET_RESULTS */;
17 /*140101 SET @OLD_COLLATION_CONNECTION=@COLLATION_CONNECTION */;
18 /*140101 SET NAMES utf8 */;
19
20 --
21 -- Database: `u122999601_toxo`
22 --
23
24 -----
25
26 --
27 -- Table structure for table `Nueva Tabla`
28 --
29
30 CREATE TABLE IF NOT EXISTS `Nueva Tabla` (
31   `Campo1` tinyint(4) NOT NULL AUTO INCREMENT,
32   `Campo2` varchar(200) COLLATE utf8_unicode_ci NOT NULL DEFAULT 'Valor por defecto',
33   `Campo3` date NOT NULL,
34   PRIMARY KEY (`Campo1`),
35   UNIQUE KEY `Campo2` (`Campo2`)
36 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
37
38 /*140101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
39 /*140101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
40 /*140101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
41
SQL Ln 1, col 1 Sel 0 (1) 1178 chars, 41 lines UNIX / OS X UTF-8 w/o BOM INS
```

Al darle al botón de IMPORT haremos el proceso inverso, cargaremos ese documento para que el gestor lo ejecute. Le damos al botón Examinar para seleccionar el archivo desde nuestro disco.

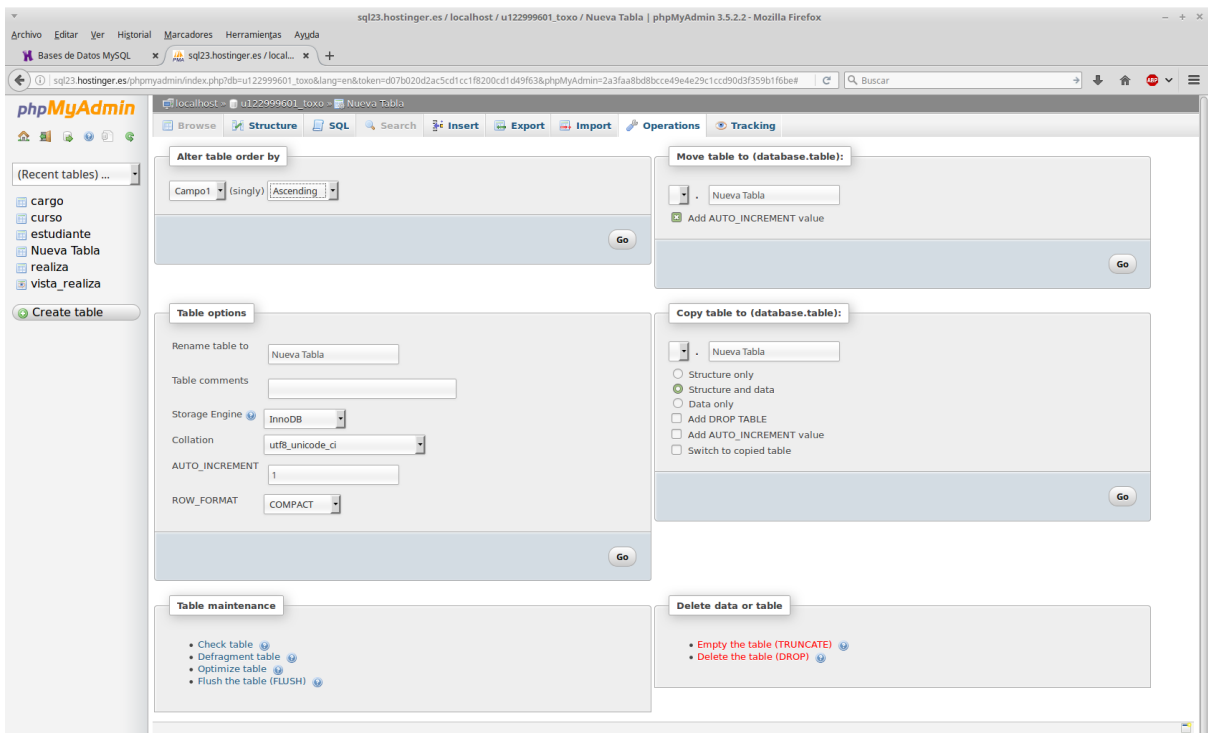


Y ahora le damos al botón GO.





En el botón OPERATIONS tenemos algunas operaciones para hacer con la tabla como cambiarle el orden de los campos, renombrar la tabla, cambiar el tipo de codificación, etc.



TRACKING, este botón es muy interesante pues podremos crear una "instantánea" o "snapshot" de las tablas que queramos y a partir de ahí irá añadiendo entradas según se vaya modificando, muy interesante si en algún momento estamos a realizar pruebas y se nos borra o corrompe la tabla. Para activarlo hay que darle al botón ACTIVATE NOW.

The screenshot shows the phpMyAdmin interface for a MySQL database. The main content area displays options for tracking changes to the table 'Nueva Tabla'. There are two main sections:

- Activate tracking for u122999601_toxo.Nueva Tabla:** This section contains an 'Activate now' button.
- Create version 2 of u122999601_toxo.Nueva Tabla:** This section allows creating a snapshot of the current table state. It includes two lists of SQL statements to track:
 - Track these data definition statements:**
 - ALTER TABLE
 - RENAME TABLE
 - CREATE TABLE
 - DROP TABLE
 - CREATE INDEX
 - DROP INDEX
 - Track these data manipulation statements:**
 - INSERT
 - UPDATE
 - DELETE
 - TRUNCATE

At the bottom right of the 'Create version' section, there is a 'Create version' button. A table at the top shows the current tracking status for the table.

Database	Table	Version	Created	Updated	Status	Show
u122999601_toxo	Nueva Tabla	1	2017-01-31 19:24:03	2017-01-31 19:24:03	not active	Tracking report Structure snapshot

Y si le damos al texto subrayado que pone TRACKING REPORT nos mostrará un pequeño historial de lo sucedido con esa tabla.

The screenshot shows the phpMyAdmin interface with the Tracking report for the table 'Nueva Tabla'. The report is titled 'Tracking report [Close]' and shows tracking statements for UPDATE, INSERT, DELETE, TRUNCATE, CREATE TABLE, ALTER TABLE, RENAME TABLE, DROP TABLE, CREATE INDEX, and DROP INDEX. The first statement is 'DROP TABLE IF EXISTS `Nueva Tabla`;', and the second is 'CREATE TABLE `Nueva Tabla` (...)' with columns 'Campo1', 'Campo2', and 'Campo3'. The table is currently active and tracking is enabled.

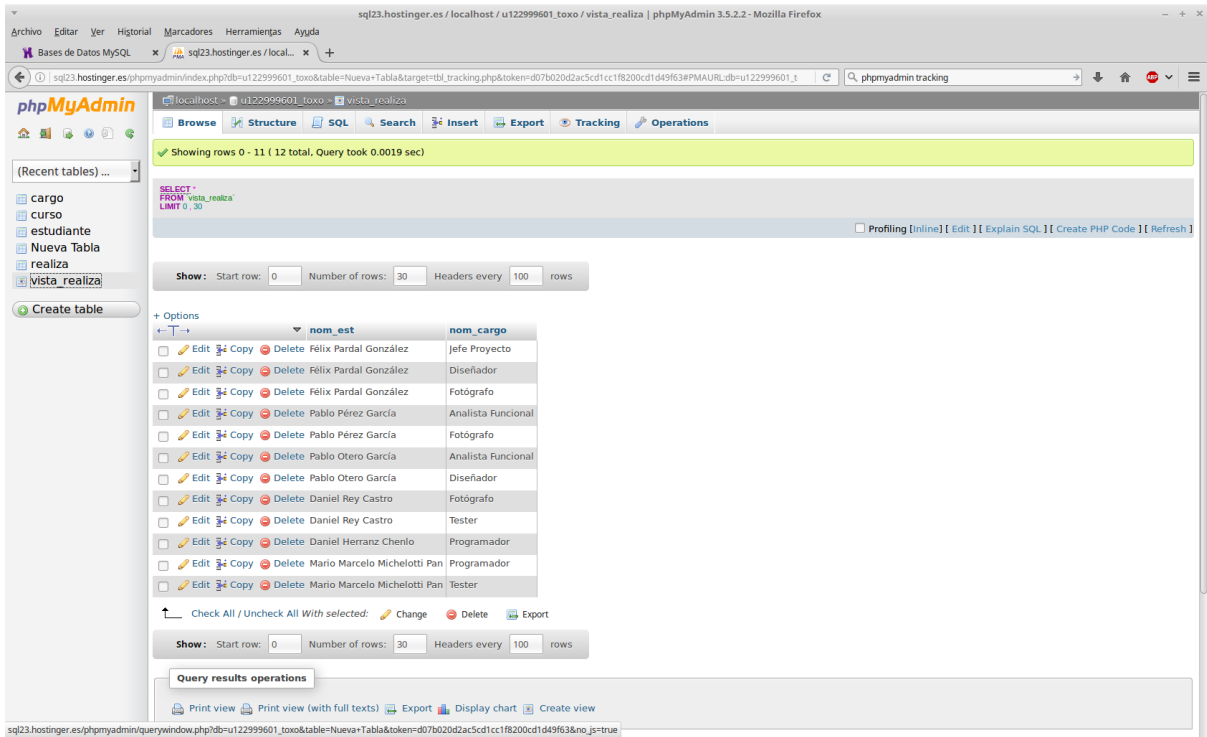
#	Date	Username	Data definition statement	Delete
1	2017-01-31 19:24:03	u122999601_toxo	DROP TABLE IF EXISTS `Nueva Tabla`;	Delete
2	2017-01-31 19:24:03	u122999601_toxo	CREATE TABLE `Nueva Tabla` ('Campo1' INT(11) 4 NOT NULL AUTO_INCREMENT 'Campo2' VARCHAR(200) COLLATE utf8_unicode_ci NOT NULL DEFAULT 'Valor por defecto', 'Campo3' DATE NOT NULL, PRIMARY KEY ('Campo1') UNIQUE KEY 'Campo2' ('Campo2') ENGINE = INNODB DEFAULT CHARSET = utf8 COLLATE = utf8_unicode_ci;	Delete

Haciendo una prueba más, se ha insertado una fila a la tabla y efectivamente, el tracking funciona perfectamente.

The screenshot shows the phpMyAdmin interface with the Tracking report for the table 'Nueva Tabla'. The report shows a new tracking statement: 'INSERT INTO `Nueva Tabla` ('Campo1', 'Campo2', 'Campo3') VALUES (NULL, 'Valor por defecto2', '2017-01-31');'. The table is still active and tracking is enabled.

#	Date	Username	Data manipulation statement	Delete
3	2017-01-31 19:29:00	u122999601_toxo	INSERT INTO `Nueva Tabla` ('Campo1', 'Campo2', 'Campo3') VALUES (NULL, 'Valor por defecto2', '2017-01-31');	Delete

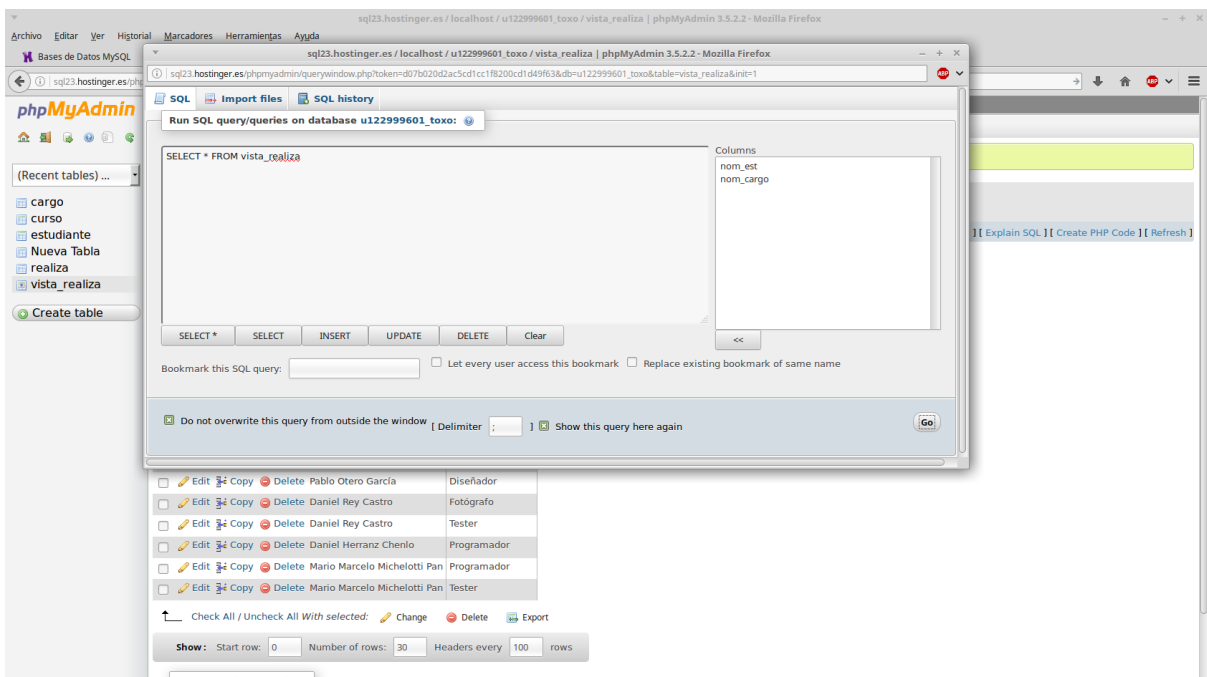
Veamos a la vista, click sobre ella.



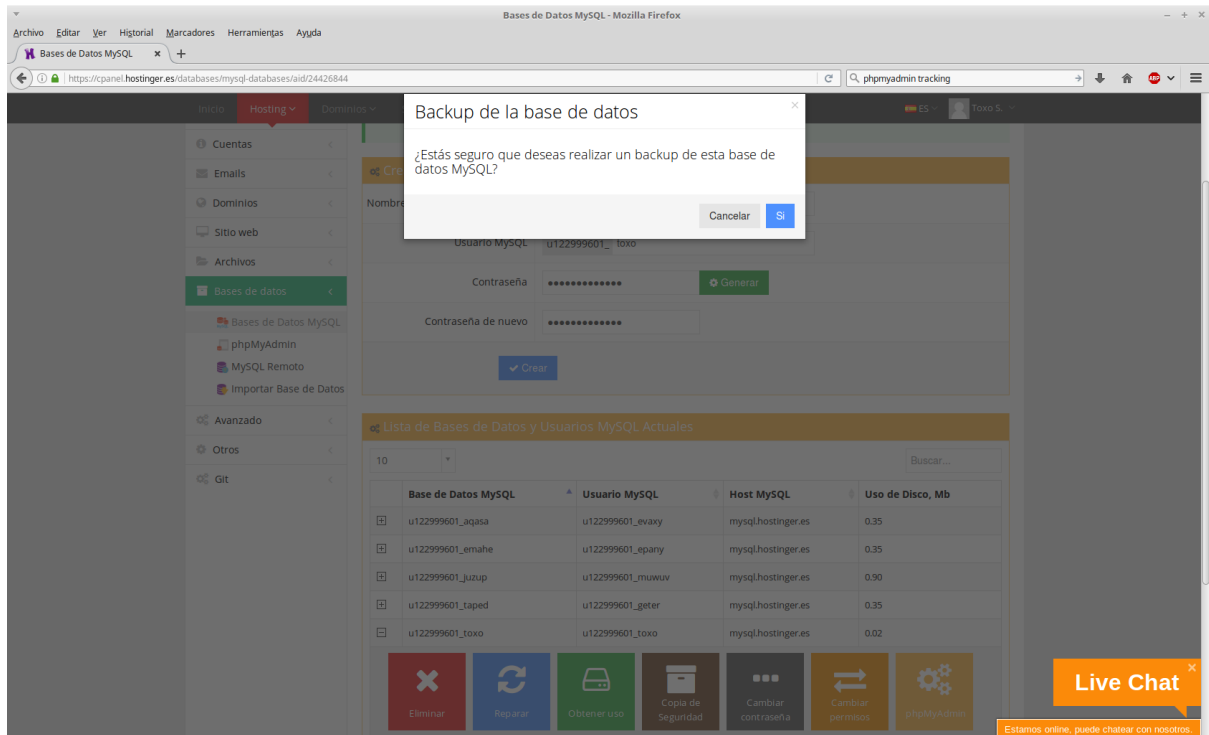
Y probaremos el ícono para ejecutar sentencias sql que está debajo del logo de phpMyAdmin. Escribimos una consulta:

```
Select * from vista_realiza;
```

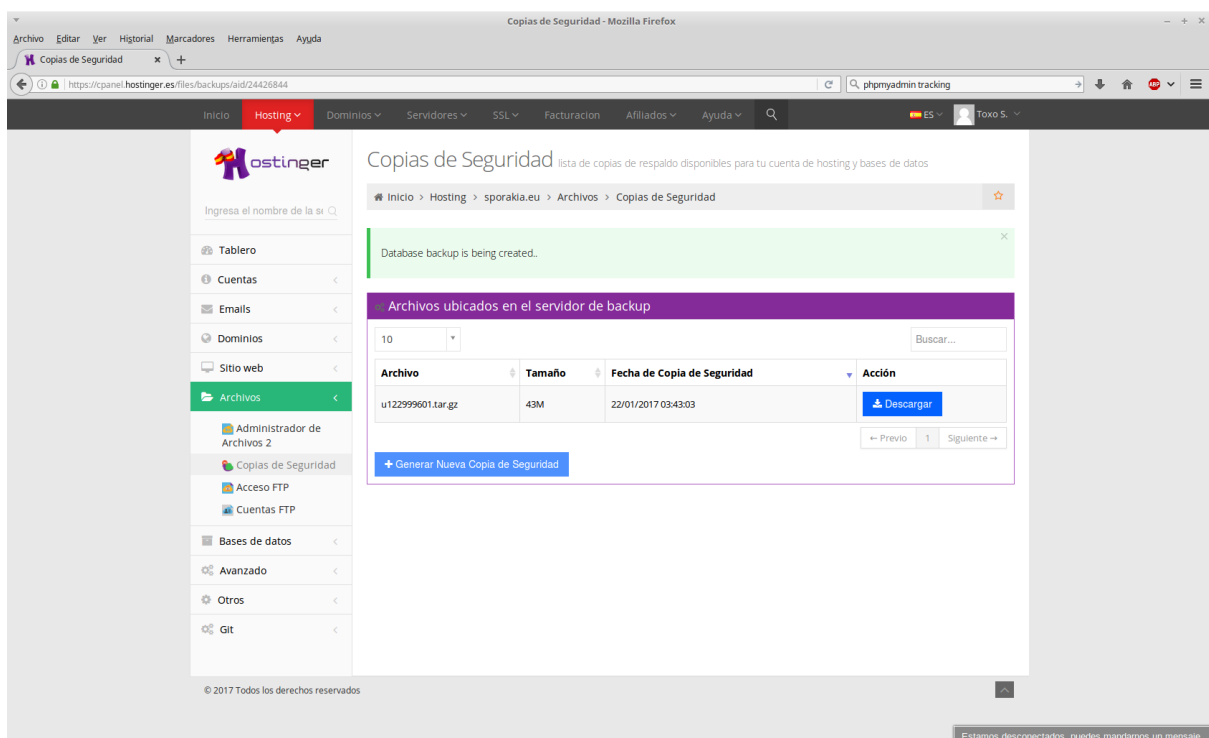
Y presionamos el botón GO. Vemos que la ventana no desaparece ni cambia y que por detrás en la otra ventana (la del navegador) aparecen los resultados, muy útil si queremos realizar una consulta tras otra.

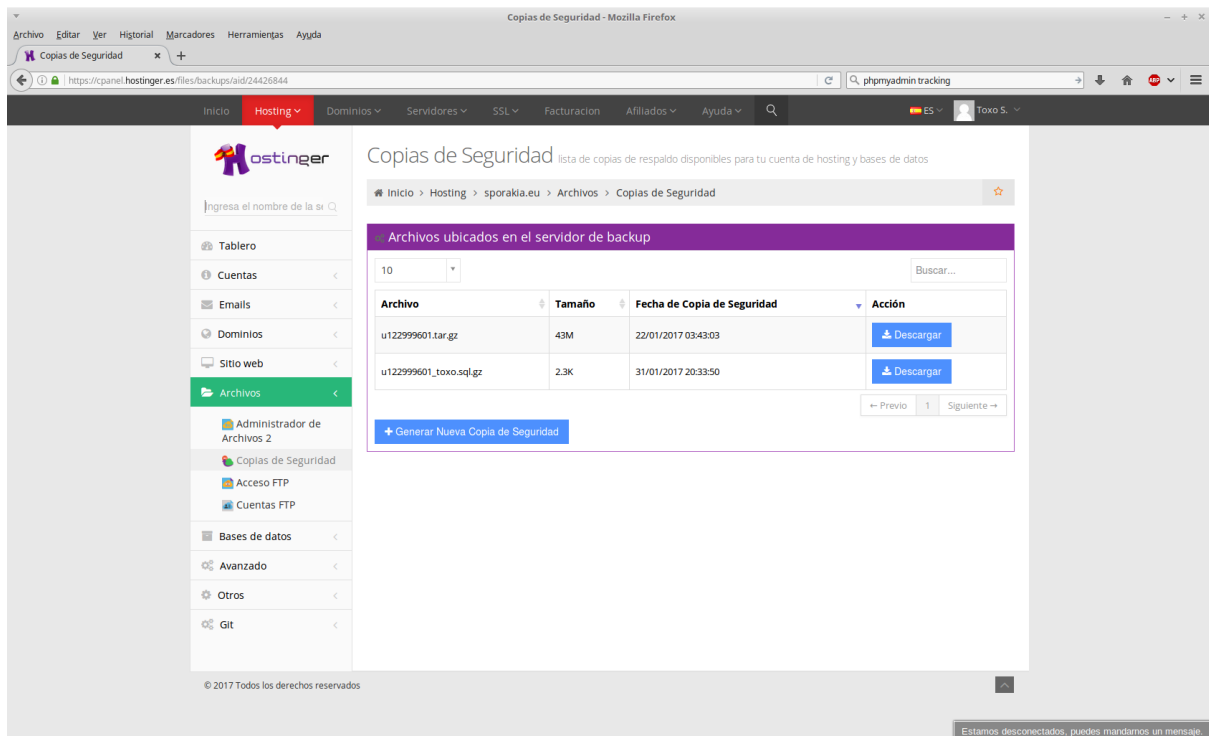


Terminamos de ver a phpMyAdmin y volvemos a la página de Hostinger donde nos habíamos quedado antes de ingresar a phpMyAdmin. Vamos a probar hacer un backup de la base de datos.

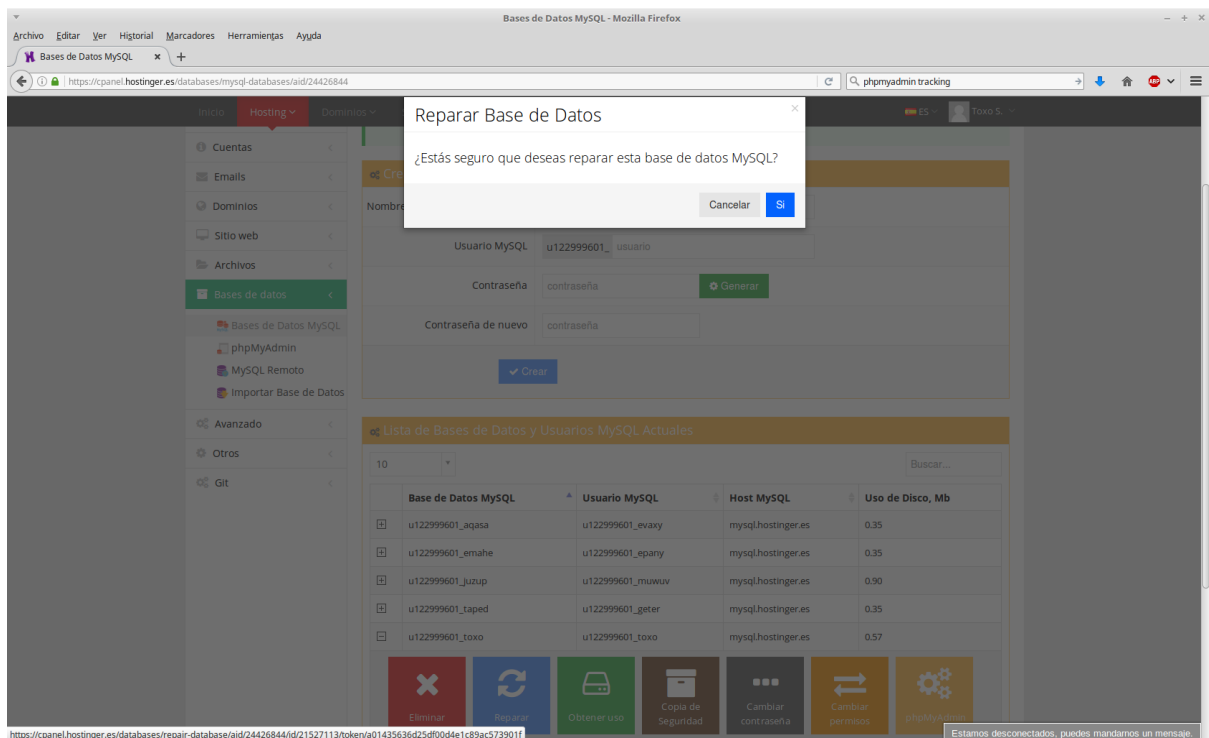


Y vemos que va todo muy fino. Unos momentos más tarde aparece también un script sql.



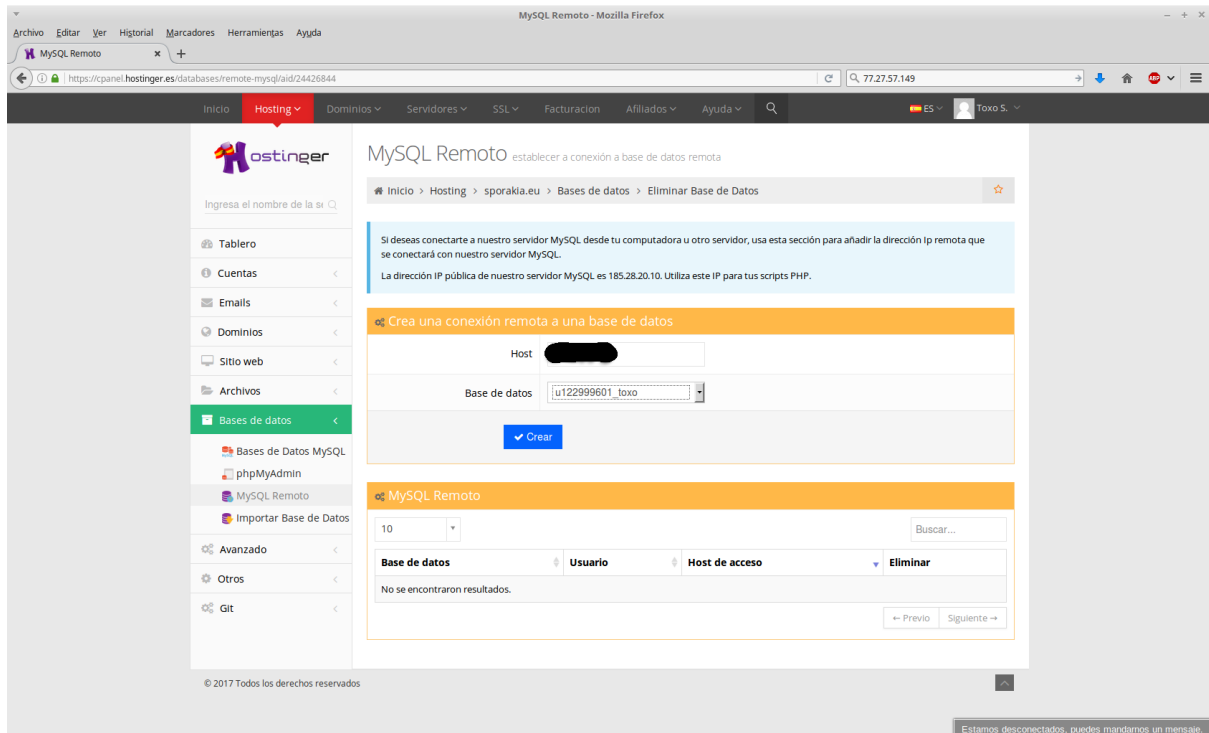


Probamos el botón de REPARAR para ver qué tal (aunque no le pase nada hay que probarlo), y sep, no corrompe ni deteriora la base de datos aunque funcione correctamente.



Vamos a probar conectarnos a la base de datos desde línea de comandos de forma remota (para aprovechar y probar todo lo que nos ofrecen). Tener en cuenta que la Dirección IP que habilitemos sea la

de nuestro router, en este caso pondré la de mi casa y seleccionaré la base de datos que creamos. En donde pone HOST, es ahí donde ingresamos nuestra IP pública.



Abrimos una terminal y metemos comandos sin añadir la contraseña, los puntos y comas suelen dar error al ponerlos directamente en el comando. Previamente tuve que instalar el paquete `mysql-server` o bien solamente el `mysql-client`, como estoy trabajando sobre entorno Linux, la conexión va muy fluida y sin problemas.

`mysql`

`-h -> HOST` de destino (la ip que nos habilita Hostinger en el mensaje de arriba)

`-u -> usuario` (el de la base de datos al momento de crearla)

`-p ->` (para que pida contraseña, lo dejamos en blanco por lo mencionado)

`<suelto y separado> ->` el nombre de la base de datos a la cual nos queremos conectar

Muestro las tablas y hago una consulta sobre la vista `vista_realiza`.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
~ $ mysql -h 185.28.20.10 -u u122999601_toxo -p u122999601_toxo
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6239960
Server version: 5.5.5-10.0.28-MariaDB MariaDB Server

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables;
+-----+
| Tables_in_u122999601_toxo |
+-----+
| Nueva Tabla                |
| cargo                      |
| curso                     |
| estudiante                 |
| realiza                    |
| vista_realiza              |
+-----+
6 rows in set (0,05 sec)

mysql> select * from vista_realiza;
+-----+-----+
| nom_est                | nom_cargo                |
+-----+-----+
| Félix Pardal González | Jefe Proyecto           |
| Félix Pardal González | Diseñador                |
| Félix Pardal González | Fotógrafo                |
| Pablo Pérez García    | Analista Funcional      |
| Pablo Pérez García    | Fotógrafo                |
| Pablo Otero García    | Analista Funcional      |
| Pablo Otero García    | Diseñador                |
| Daniel Rey Castro     | Fotógrafo                |
| Daniel Rey Castro     | Tester                   |
| Daniel Herranz Chenlo | Programador              |
| Mario Marcelo Michelotti Pan | Programador              |
| Mario Marcelo Michelotti Pan | Tester                   |
+-----+-----+
12 rows in set (1,11 sec)

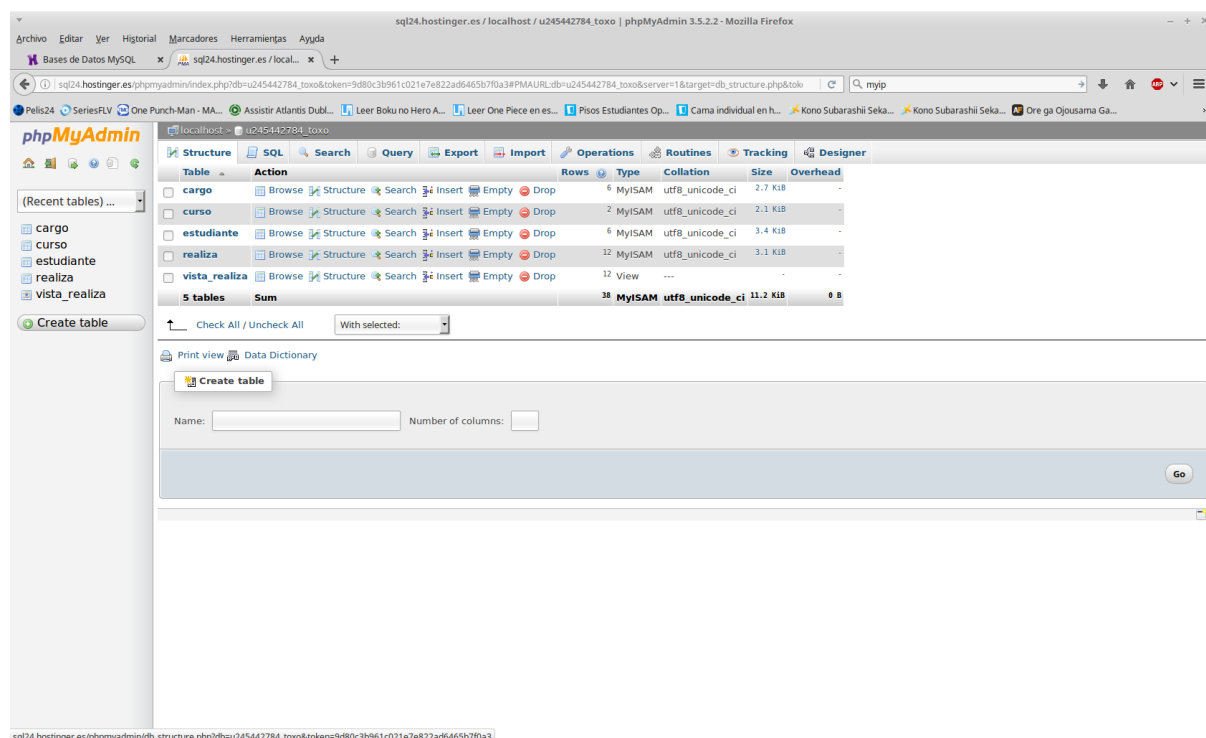
mysql> █
```

Finalizadas las pruebas, hemos comprobado cómo utilizar phpMyAdmin y a conectarnos mediante línea de comandos.

Diferencias entre cuentas de Hostinger

Existen algunas diferencias entre las cuentas de hostinger.es al momento de trabajar con bases de datos, la 1ra es muy razonable: no es posible conectarse desde el exterior. La 2da diferencia es en el comportamiento al momento de crear una base de datos, utilizando el mismo script sql sin modificar absolutamente nada, no me ha creado las claves foráneas, un tema muy serio para conservar la integridad de los datos, pero por suerte para todo hay una solución, vincular las tablas mediante el entorno de phpMyAdmin.

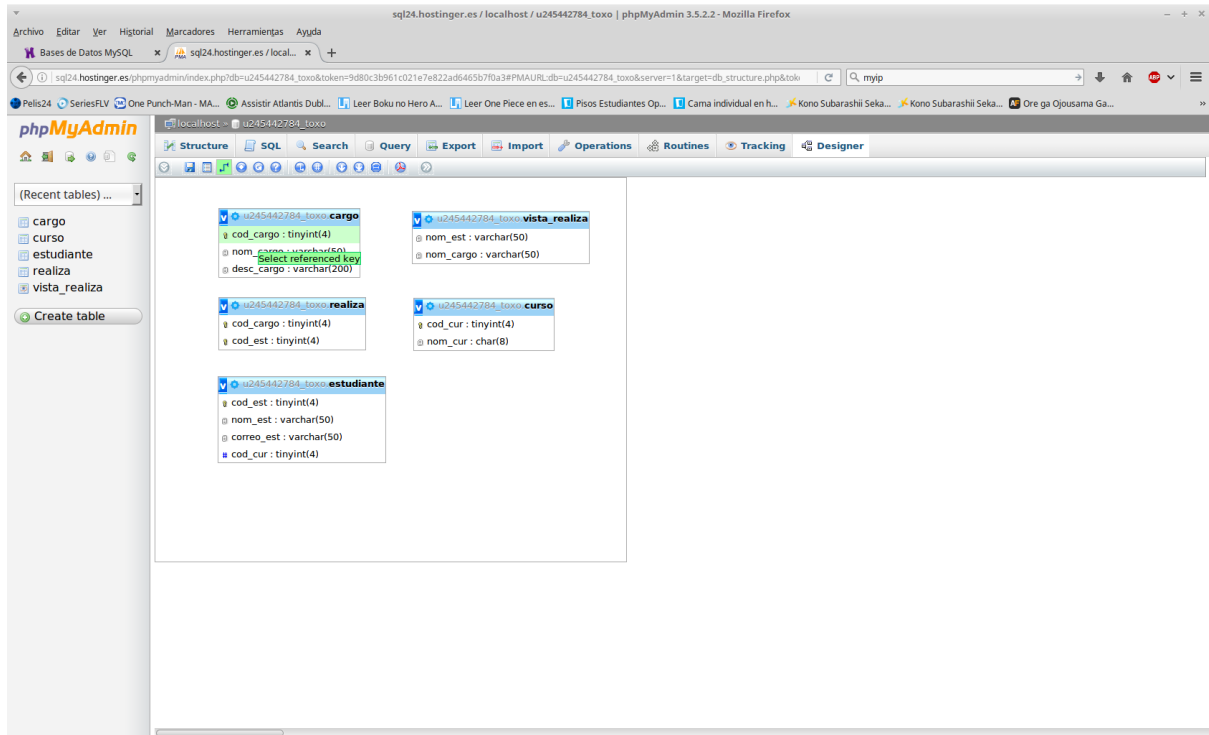
1ro, entramos en phpMyAdmin y clicamos sobre el botón que parece una casita, ubicado debajo del logo a la izquierda del todo.



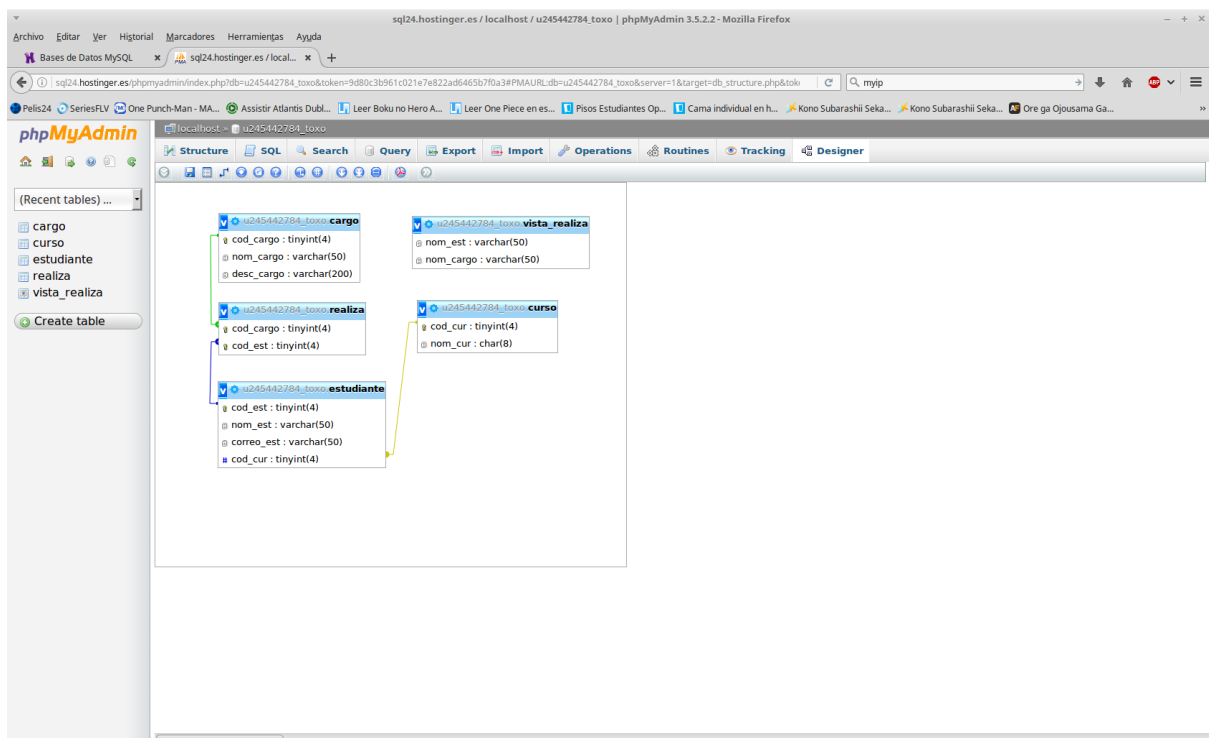
The screenshot shows the phpMyAdmin interface in a browser window. The main area displays a table structure for a database. Below the table structure, there is a 'Create table' dialog box with a 'Name:' field and a 'Number of columns:' field. The 'Designer' menu item is visible in the top navigation bar.

Table	Action	Rows	Type	Collation	Size	Overhead
cargo	Browse Structure Search Insert Empty Drop	6	MyISAM	utf8_unicode_ci	2.7 KIB	-
curso	Browse Structure Search Insert Empty Drop	2	MyISAM	utf8_unicode_ci	2.1 KIB	-
estudiante	Browse Structure Search Insert Empty Drop	6	MyISAM	utf8_unicode_ci	3.4 KIB	-
realiza	Browse Structure Search Insert Empty Drop	12	MyISAM	utf8_unicode_ci	3.1 KIB	-
vista_realiza	Browse Structure Search Insert Empty Drop	12	View	---	-	-
5 tables	Sum	38	MyISAM	utf8_unicode_ci	11.2 KIB	0 B

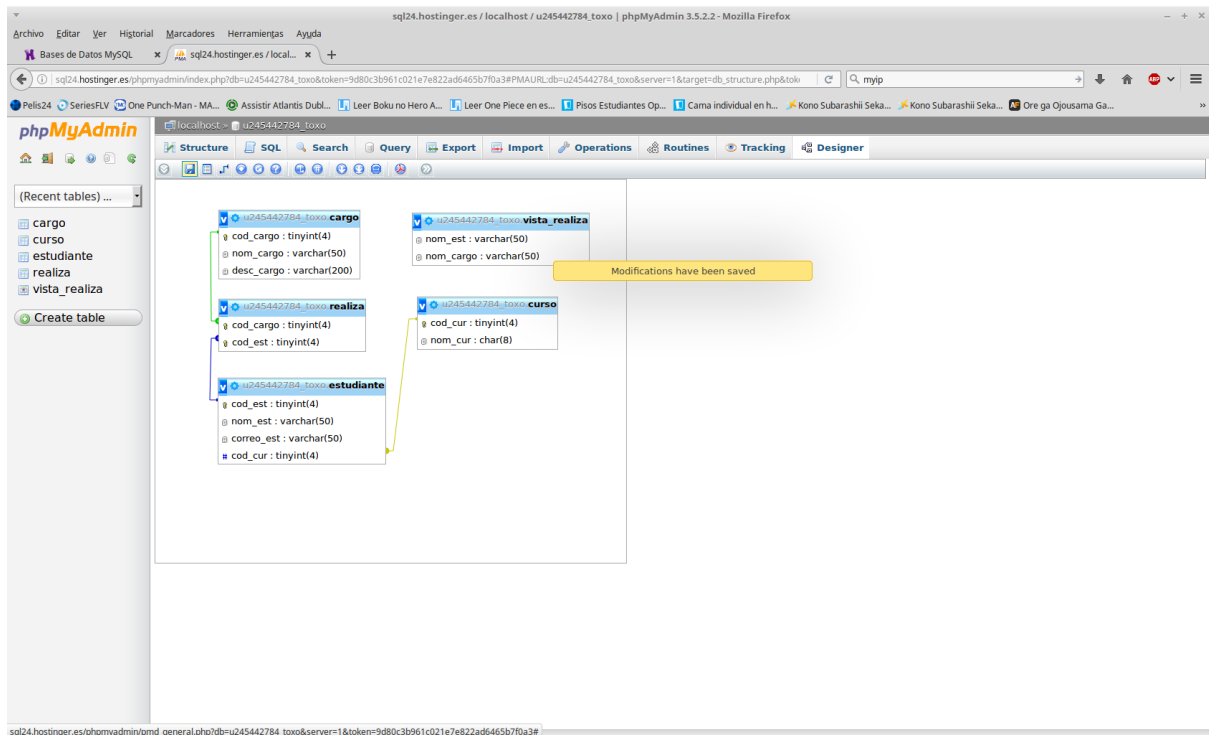
2do, En la barra de menús, aparece al último uno que pone "Designer", clicamos sobre él y veremos las tablas sueltas sin conexión alguna entre ellas (foreign keys), para que se viera mejor me tomé el tiempo de ordenarlas según el modelo entidad relación. Ahora clicamos sobre el ícono número 4 (una S cuadrada con una flecha al final), veremos cómo aparece un recuadro verde indicando qué tenemos que seleccionar y clicamos sobre la llave primaria de la tabla cargo (cod_cargo). Veremos cómo aparece un recuadro verde indicando qué tenemos que seleccionar.



Ahora clicamos sobre la llave el campo correspondiente a la siguiente tabla (realiza en este caso) para que vincule esa primary key con la siguiente primary key, creando un enlace de referencia. Confirmamos el cuadrado de diálogo y luego de completar todos los enlaces se tendría que ver como algo así.



Guardamos clicando en el 2do ícono (un diskette), y listo, ya tenemos las claves foráneas funcionando.



Preparar la conexión

Para ir preparando la conexión, tendremos que tomar datos relevantes como nombre de usuario, contraseña, base de datos y el host.

En este caso utilizaremos el host interno de hostinger.es, porque la conexión no saldrá de sus infraestructuras, quedando los datos de la siguiente manera:

Host → `mysql.hostinger.es`

User → `*****`

Pass → `*****`

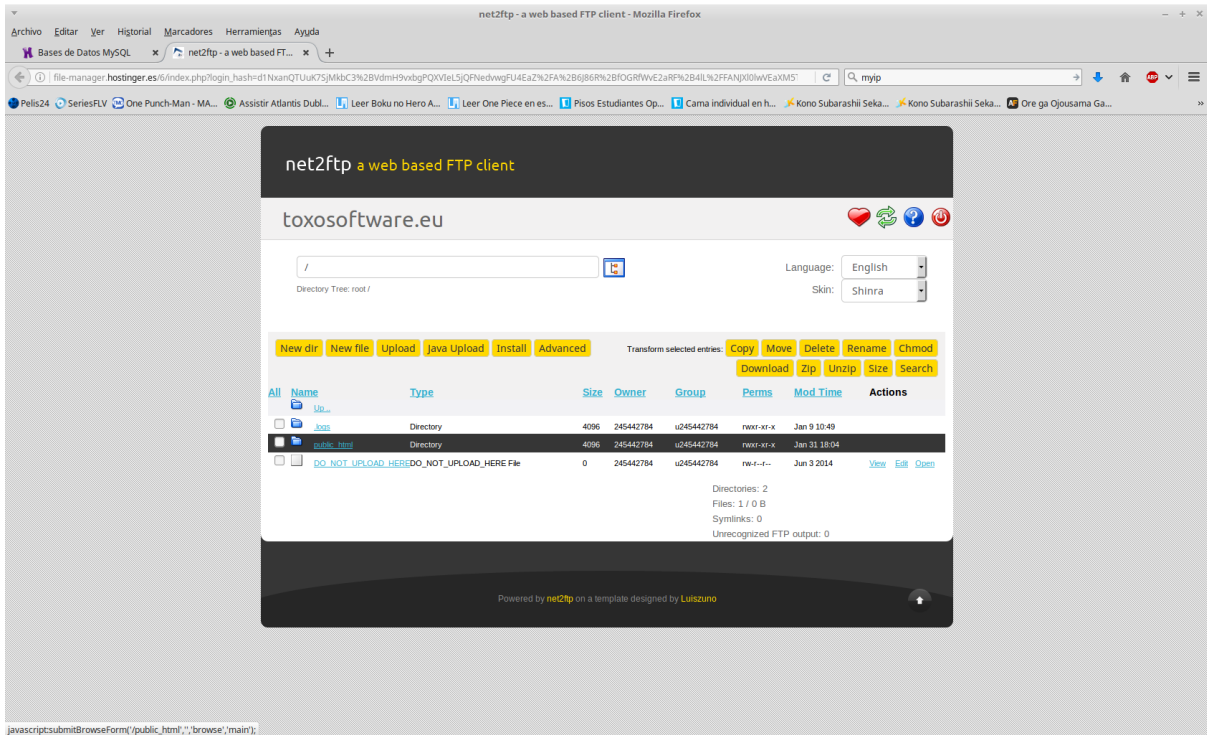
DB → `u245442784_toxo`

Aquí una vista de la base de datos creada.

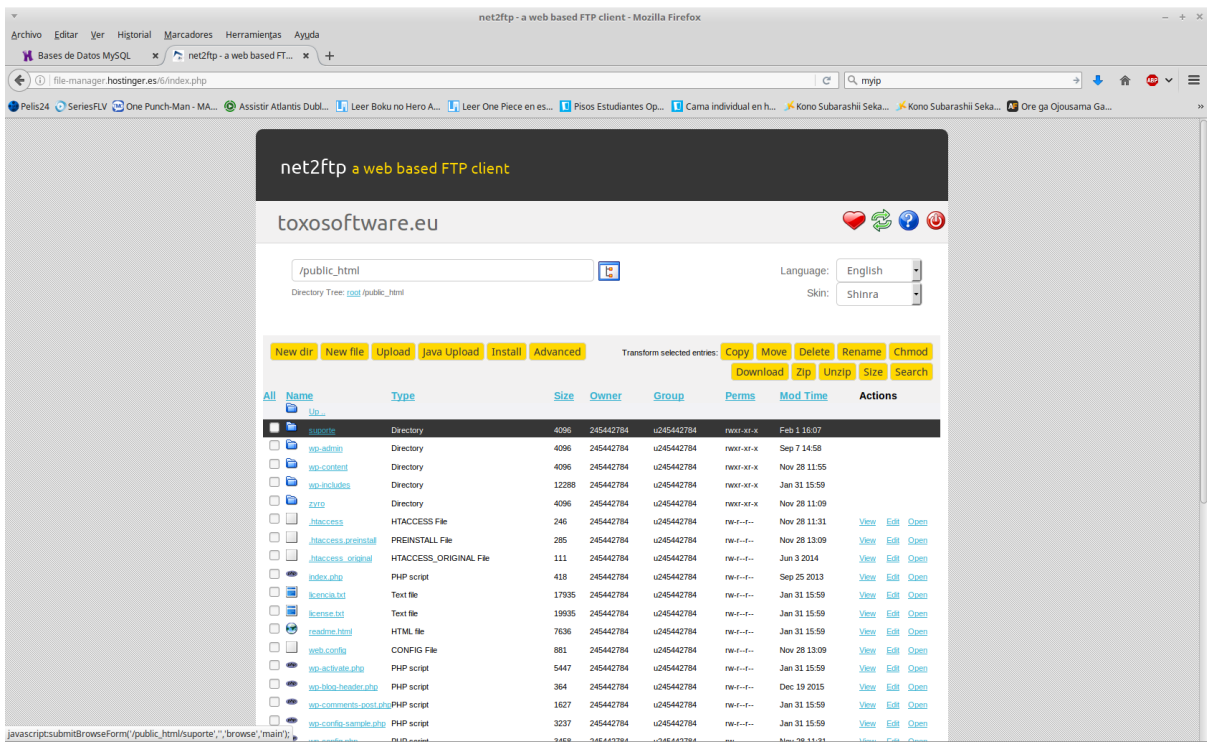
© 2017. Todos los derechos reservados
<https://cpanel.hostinger.es/files/manager-2/aid/24496177>

Estamos online, puede chatear con nosotros

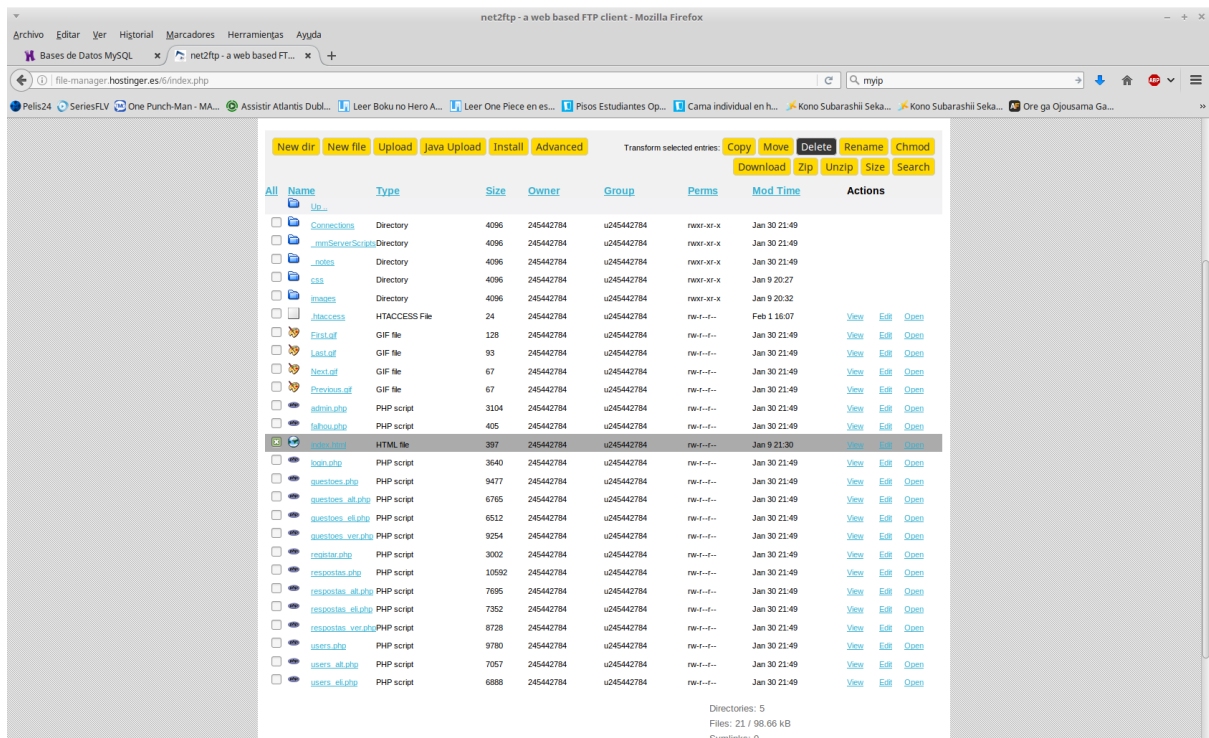
Ahora iremos a cambiar el viejo archivo `index.html` por el nuevo `index.php`, para que el servidor web reconozca automáticamente que dentro hay también código php y poder interpretarlo. Accedemos al panel de Administrador de Archivos 2, a la izquierda de la página. Nos debería aparecer una especie de navegador de ficheros con el nombre `net2ftp` en la cabecera y el sitio del ftp `toxosoftware.eu`. Hacemos click en la carpeta `public_html`.



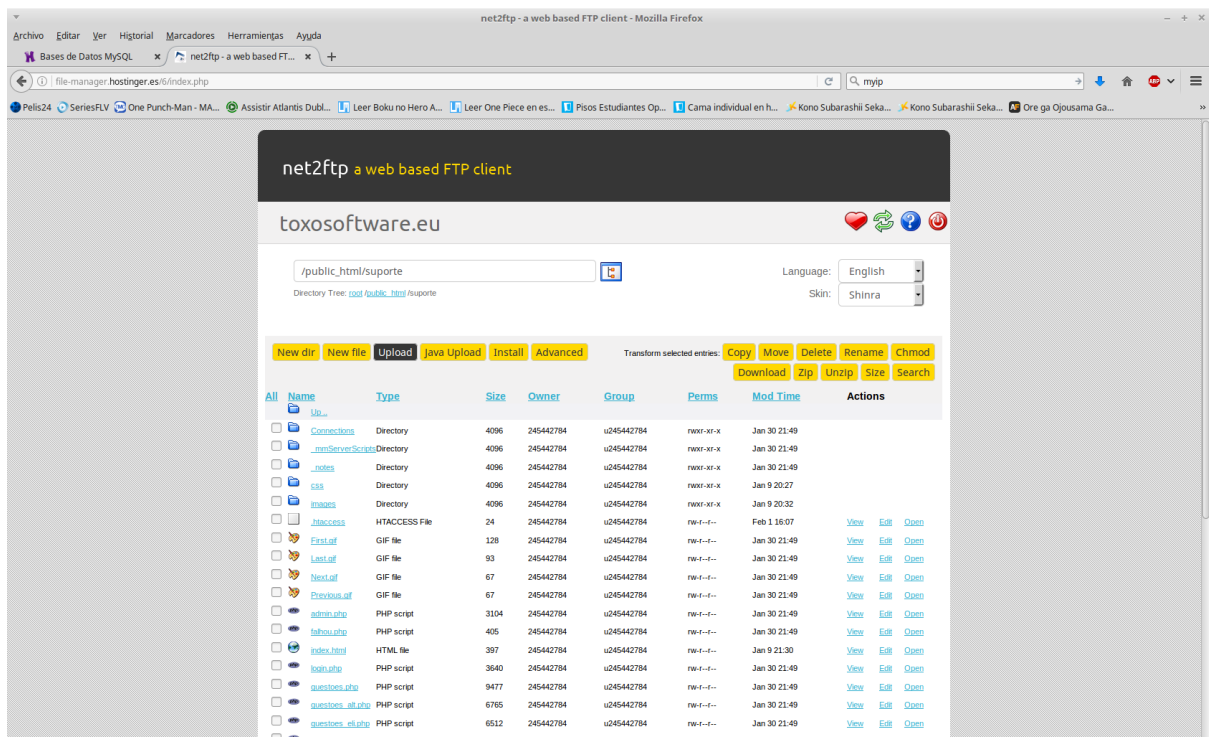
Click en soporte donde tenemos actualmente la página en construcción, pero para hacer pruebas nos vale de sobra.



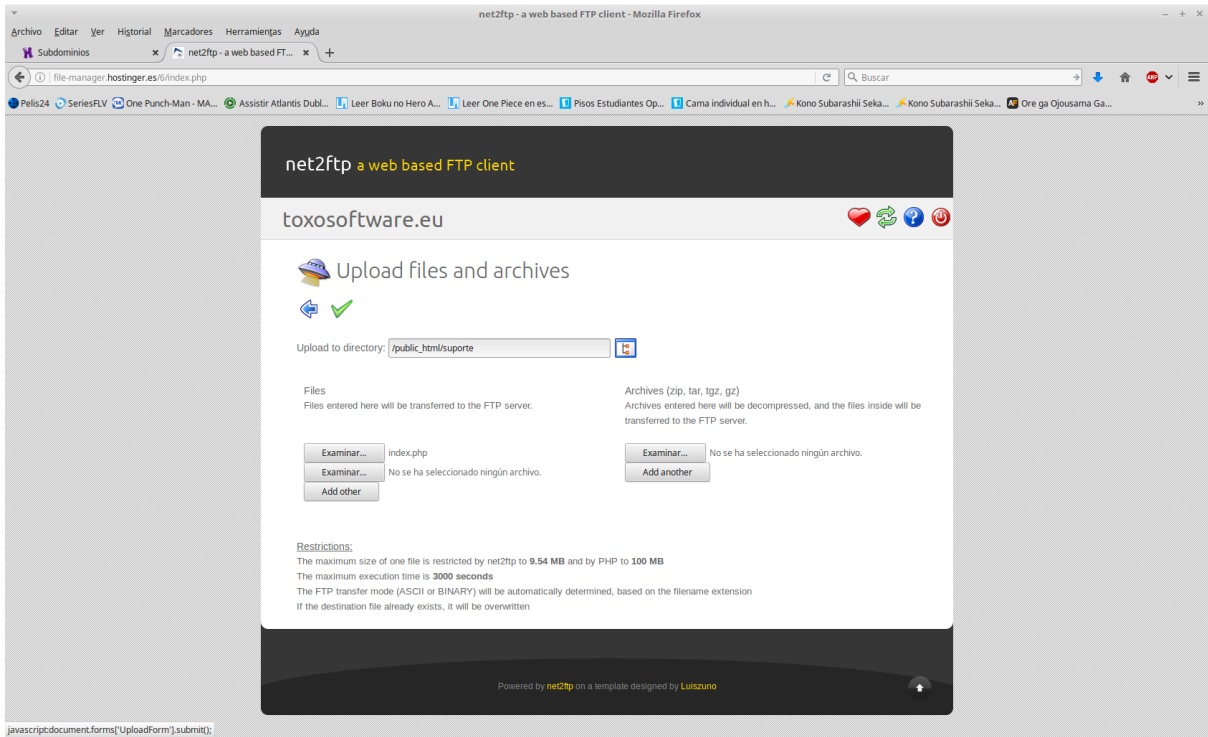
Eliminamos el archivo viejo index.html marcándolo y luego clicando en el botón "Delete".



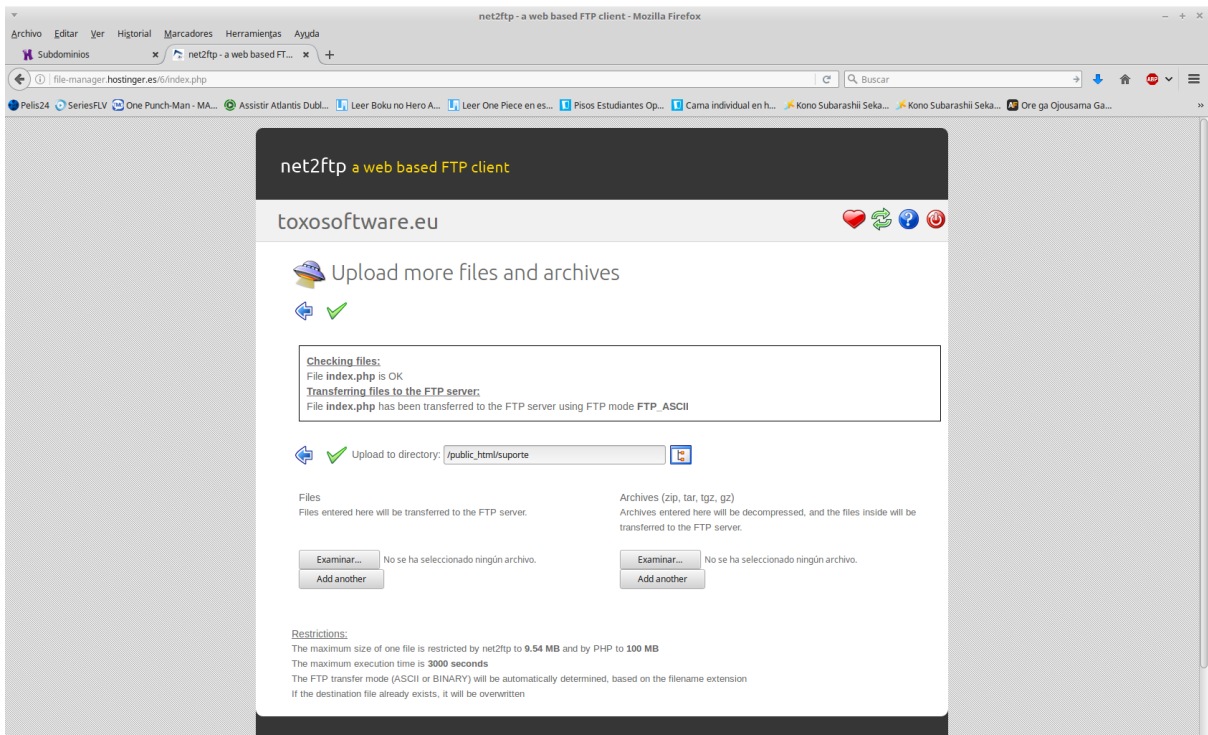
Ahora clicamos en el botón de "Upload", para subir el nuevo archivo.



Clicamos en el 1er "Examinar" de la izquierda, que está pensado para subir archivos, seleccionamos el archivo correcto y luego le damos al "tick" verde de arriba que está justo debajo del ovni.



Listo, nos informa que el archivo se cargó exitosamente.



Vamos a ver los subdominios.

Subdominios - Mozilla Firefox

Subdominios | net2ftp - a web based FT... | En construcción

https://cpanel.hostinger.es/domains/subdomains/aid/24496177

Inicio | **Hosting** | Dominios | Servidores | SSL | Facturacion | Afiliados | Ayuda

Subdominios crear y administrar subdominios

Inicia > Hosting > toxosoftware.eu > Dominios > Subdominios

Crear Nuevo Subdominio

Ingresar subdominio .toxosoftware.eu

Lista de Subdominios Actuales

10

Subdominio	Directorio	Acciones
soporte.toxosoftware.eu	/home/u245442784/public_html/soporte	<input type="button" value="Eliminar"/>

← Previo 1 Siguiente →

Live Chat

© 2017 Todos los derechos reservados Estamos online, puede chatear con nosotros

Código PHP

Este es el código que añadí al final de la página web para probar la conexión hacia la base de datos.

NOTA: el método procedural ha quedado obsoleto y en [hostinger.es](https://www.hostinger.es) recomiendan el método orientado a objetos, por esa razón y porque para mi es muchísimo más cómodo, utilizaré el 2do.

```
<?php
// Datos de la conexión
$host = "mysql.hostinger.es";
$user = "*****";
$pass = "*****";
$dbname = "u245442784_toxo";
// Nuevo objeto para la conexión
$mysql = new mysqli ($host,$user,$pass,$dbname);
// Comprobar si la conexión da error
if ( $mysql -> connect_error ) {
    die("No che puede: ". $mysql -> connect_error);
}
// Cambiar el charset a utf-8
$mysql -> query ("SET character_set_results = 'utf8',
character_set_client = 'utf8', character_set_connection =
'utf8', character_set_database = 'utf8', character_set_server
= 'utf8'");
// Consulta de prueba
$resultado = $mysql -> query("select * from vista_realiza");
if( $mysql -> connect_error ) {
    die ("No che puede: ". $mysql -> connect_error);
}
if ($resultado -> num_rows > 0) {
    echo "Estudiante          -      Cargo<br><br>";
    while ($fila = $resultado -> fetch_assoc()) {
        echo $fila['nom_est']."          -
".$fila['nom_cargo']."<br>";
    }
}
$mysql -> close();
?>
```

Ahora toca ver los resultados, debería mostrar al final de la página, una impresión de cada una de las filas que hay dentro de la tabla vista_realiza, que muestra el nombre del estudiante y su cargo en la empresa.



Prueba concluida. Todo funciona a la perfección.

NOTA: en el código incluí un par de líneas para comprobar a modo de aprendizaje los posibles errores que pudieran ir apareciendo, esto NUNCA se debe de realizar en la realidad, es preferible mostrar un mensaje que diga: "Estamos experimentando algunas incidencias, por favor disculpe las molestias y vuelva más tarde, gracias...", o algo por el estilo. Es muy perjudicial mostrar datos relevantes y/o sensibles de la empresa.